Development of an FPGA and MCU based Stack-able Processing platform incorporated with on-board compute module for Real-time processing applications

by

©Teng Wang

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

Master of Engineering

Department of Engineering and Applied Science

Memorial University of Newfoundland

November 2017

St. John's Newfoundland

Abstract

The focus of this thesis is to develop an FPGA and MCU-based stackable processing platform incorporated with an on-board computer module for real-time processing applications. The goal is to deliver a compact-sized hardware platform with extensible capabilities to provide high-speed, parallel computing with low power consumption. This hardware platform is named ioNeurons and consists of three module types: processing modules, sensing modules, and interface modules. The ioNeurons ecosystem design is based on combining individual strengths into highly adaptable and powerful solutions. The processing modules are stackable in no particular order, allowing the ability to match multiple modules' individual capabilities to the project's needs. Developers can assign tasks to multiple processing modules according to the different real-time requirements. The implementation of a small-scale quadrotor helicopter is introduced as an application of this hardware platform.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Nicholas Krouglicof for providing me this great opportunity to work on and develop such interesting and rewarding projects. With his outstanding guidance and support I learned and believes no detail is insignificant when designing. His guidance benefited me in the past, now and the rest of my life.

I would like to thank my parents, Mrs. Ping Li and Mr. Yanzhou Wang for their invaluable support, love and encouragement.

I would like to thank my wife, Mrs. Xin Huang. Without her continued understanding, moral support and love, this thesis would not have been possible.

I would like to thank Dr. Nicholas Krouglicof and Dr. Ralf Bachmayer for providing me a full time job which related to my researches and interest in Memorial University as a Research and Project Engineer between 2013 to 2016. I learned how to work with colleagues for a common goal, to be a qualified contributor in a harmonious team. Challenges in experiments, problems in a practical application, all those obstacles makes me stronger and smarter. When problems are solved, when we move forward even a small step, all those small things power me to go further. Even in a hard time, I won't forget the encouraged smiles were on their faces.

Facing difficulties, I believe that you have one less reason to hang back, when there is one more friend who unswervingly trusts you. Thanks for every friend who trusts

me. Special thanks to my peers Mohammed Raju Hossain, Stephen Reddin, Michael Morgan, Dennis Fifield.

Lastly, I dedicate this thesis to my children Ayden Wang, and Olivia Wang. I love you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Literature Review

1.1.1 FPGA

A field programmable gate array (FPGA) is a kind of integrated circuit (IC) that can be electrically programmed using a hardware description language (HDL) to become almost any kind of digital circuit. In 1984, Altera offered its first programmable-logic chip, EP300 [3], and in 1985 Xilinx co-founders Ross Freeman and Bernard Vonderschmitt invented the first commercially available field programmable gate array, XC2064, which offered 800 gates [4]. FPGA technology has experienced rapid development. Today, the highest density FPGA fabric can have up to 5.5 million logic elements and FPGAs have become a popular implementation media for digital circuits. The development of manufacturing process technology and the continuous improvement of FPGA routing architectures have significantly improved the ability of FPGAs to handle logic operations, making it possible to implement large-scale and complex logic circuits. At the same time, embedded system developers use FPGAs to achieve more complex, large-scale designs based on project requirements. FPGAs provide solutions for embedded designs that offer faster processing speeds and lower power consumption [1].

In an FPGA architecture, there are three component parts: configurable logic blocks (CLBs), programmable routings, and I/O blocks. The configurable programmable logic block is a basic logic and storage component which is used to implement logic functions. The programmable routings are used to connect these logic functions, and the I/O blocks connect the logic blocks to off-chip connections through routing interconnects [1]. Figure 1.1 shows a generalized example of FPGA architecture.

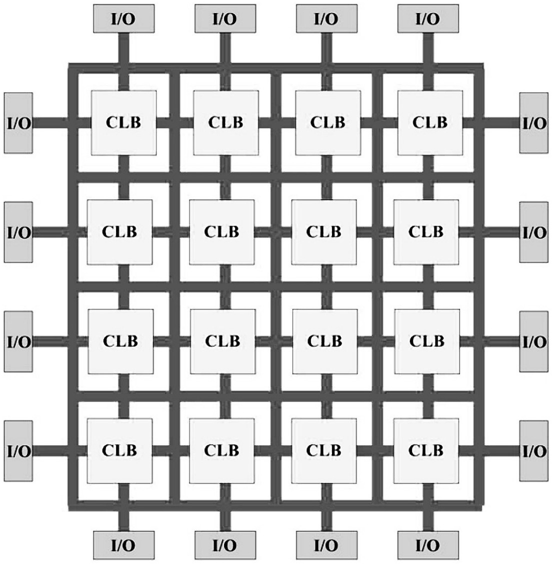


Figure 1.1: Overview of FPGA architecture [1]

Compared with Application Specific Integrated Circuits (ASICs), the most considerable advantage of FPGAs is that they provide a cheaper solution and faster time to market for low to medium volume productions. Once the ASIC is fabricated, its functions cannot be modified, but the FPGA can be re-programmed which demonstrates the flexible nature of FPGAs. However, this flexibility is at the expense of speed and size. About 90% of the internal area of FPGAs is the programmable routing interconnects. The flexible nature makes FPGAs larger, slower and more power consuming than ASICs [1]. With the continuous development of FPGA technology, this gap is gradually getting smaller. Compared to microcontrollers, FPGAs still have irreplaceable advantages.

1.1.2 MCU

It has been more than 30 years since the first microcontroller TMS1000 was introduced in 1974. Initially, they were used for applications such as calculators, cash registers, watches and measuring instruments. The Intel 8048 was the first microcontroller that became widespread. Nowadays, billions of products are made every year which contain microcontrollers; they are found throughout daily life, industry, and science, in products including household appliances, telecommunication equipment, and the automotive industry. [2]. Figure 1.2 shows the block diagram of a typical microcontroller.

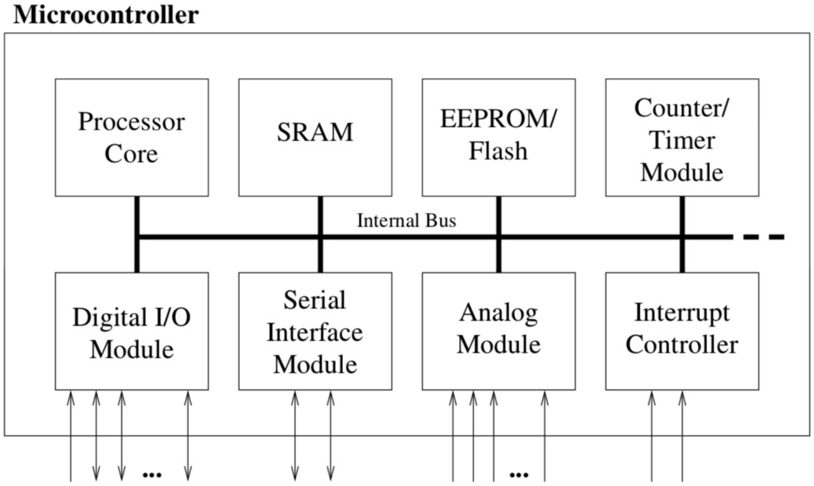


Figure 1.2: Overview of MCU architecture [2]

A microcontroller is a single integrated circuit that, at a minimum, contains a processor, memory, a clock oscillator and input & output interfaces. These are the fundamental elements of a complete computer system. All elements are connected via an internal bus [2] [5].

The processor core is the main part of a microcontroller and consists of an arithmetic logic unit, a control unit, and registers. Its primary function is to run the programme written in memory to perform specific tasks. The memory is an important part that provides storage for the software and data in embedded systems. It can be divided into two categories: non-volatile and volatile memory. The first type, non-volatile memory can retain its contents when power is removed. The Electrically Erasable Programmable Read Only Memory (EEPROM) is a kind of non-volatile memory which is used to save the complete program or an initialization routine. The second type, volatile memory, consists of Random Access Memory (RAM) which stores global variables and loses its entire contents when power is removed [2] [5] [6].

Microcontrollers execute the main program in the normal flow without any concerns, but sometimes higher priority external or internal events must be processed when an interrupt condition occurs. The interrupt controller takes the responsibility for this type of work to interrupt the normal flow operation and go to execute the prioritized task. As soon as the interrupt event is completed, the microcontroller comes back to the main program again and starts to run at the point that was interrupted [2].

The timer is, strictly speaking, a counter module which is another important part of every microcontroller. Most microcontrollers have at least one or more counters to capture and count events and measure intervals. The counter can also be used to trigger interrupts after a certain number of clock cycles, and generate pulse-width modulated (PWM) signals to control motor speed. In default mode, the timer is incremented by the system clock which can be an internal resistor-capacitor oscillator or a more accurate external quartz crystal oscillator. [2] [5].

The Input/Output (I/O) are divided into digital I/O and analog I/O, and both of them can directly interact with peripheral equipment. The I/Os are the windows of a microcontroller to the outside world. Most I/Os are bidirectional, and can be used as an input and an output. The digital I/Os have two voltage states: logic HIGH and logic LOW. Apart from their digital I/O capabilities, most of them also have many alternate functions to provide more features and save pins. While digital signals on digital I/O take discrete values, analog signals on the analog I/O are the time-varying continuous signals. When using an analog sensor and reading the signal coming from it, the microcontroller must handle an analog signal that changes continuously with time. Analog signals are sampled by Analog Digital Converters (ADCs) through the analog I/Os, and finally converted to digital values inside the microprocessor [2].

1.1.3 Applications of FPGAs and MCUs

Both FPGAs and MCUs have their irreplaceable advantages. In the past research and development, engineers and developers have used them together to implement excellent applications.

Mohammed Raju Hossain designed and implemented a high-performing quadrotor [7], and its flight controller contains an FPGA and an MCU. A proportional-integral-derivative (PID) technology-based closed loop motor speed controller was also implemented in the same FPGA-based flight controller for precise speed control of the motors. The flight controller combines the powerful computing power of an FPGA and an MCU; however, it is difficult to add other peripherals and sensors to the flight controller because of its integration with the quadrotor frame.

As another example of the processing power of FPGAs, consider the following image processing application. A recent research-based project from Université Paris-Est implemented an IMU-aided image stacking algorithm in a digital camera for Unmanned Aerial Vehicles [8]. The aim of the project is to be able to acquire several images with short exposure time and use an image processing algorithm to produce a stacked image with an equivalent long exposure time. French National Mapping Agency designed the camera that Université Paris-Est used. The camera contains an FPGA and two ARM cortex A9 processors which is a dedicated design to provide powerful real-time image processing ability in a small device, usually such a device is expensive. In the future, they will implement a real-time version by accelerating the image resampling part in the FPGA.

Nicholas Krouglicof presented a method for rigid-body pose measurement from a single perspective view in 1993 [9]. In 2012, the thesis author Teng Wang implemented a computationally efficient contour tracking algorithm [10] with FPGAs; this algorithm can obtain the shape features and generalized moments which are needed for the next step calculation of pose estimation when completing the edge detection of objects. The FPGA can process 27 frames per second (FPS) with image resolution 1024 x 768. Then the MCU used the information provided by the FPGA to complete the rest of the calculations and obtained the pose information of the object. However, at that time the MCU was Atmel's MEGA32U4 8-bit Microcontroller which provides up to 16 MIPS throughput at 16 MHz. This processing speed cannot complete calculations for all 27 frames in a second. Its output is only 5 FPS. This example shows that, although a powerful FPGA is used for image processing, it is still difficult to obtain an ideal processing speed without support from a powerful MCU. The processing power of a system often depends on the weakest part.

Salcic described a microcontroller/FPGA-based prototyping system for embedded applications [11], which proposes an embedded system to make up for the drawbacks that a computer system has in satisfying specific functions of an application. The embedded system combines a standard microcontroller and an FPGA to come up with a new solution in prototyping systems that is called PROTOS. This environment allows the development of hardware and software, efficiently and flexibly, in embedded applications. Using an FPGA to implement the complicated digital system is one of the most popular methods; as well as using an FPGA as a coprocessor to speed up the processing time in some applications in which time is a critical factor [11] [12] [13]. In Salcic's paper, Motorola 68HC11 MCUs and FLEX 8000 FPGAs are used to implement the application. In the system, all other components are coordinated by the 68HC11 MCU that runs in expanded bus mode to deal with external memories and the FPGA. However, the FPGA is dynamically reconfigurable and can be programmed to have different functions as needed in various applications.

Most of the existing computing platforms on the market only contain one type of processing chip. For examples, Arduino and ARM mbed development boards only use MCU chips. And the Raspberry Pi platform only uses SoC chip. If developers need to use FPGAs and work with MUCs to complete tasks, developers usually need to establish the connections by themselves. A US company, Adafruit, developed the MOJO FPGA development board, which includes a Spartan 6 XC6SLX9 FPGA chip, and an ATmega32U4 microcontroller on the same board, however, the microcontroller is only used to configure the FPGA chip, USB communication and reading the analog pins, it cannot really participate in the computing work with the FPGA chip. Alorium Technology company developed an XLR8 FPGA-based Arduino compatible development board. It provides the ability to implement custom logic that accelerates specific functionality that is slow on a microcontroller. However, it only provides 8K logic gates, and cannot extend the number of logic gates through the stack structure.

1.2 Motivation and Scope of Work

In the research and development of electronic robots or mobile devices, developers usually use computing platforms composed of FPGAs, MCUs, or SoCs. Choosing to design a customized computing platform on its own or purchase an existing development platform from the market is usually the first problem developers face.

Comparing the customized development platform and general development platform purchased from the market, they have their own advantages and disadvantages. The customized development platform can meet the requirements of developers in terms of size, performance and power consumption. However, graduate students and developers need to spend a lot of time and energy to complete the preparatory work for hardware design and create a hardware development platform that meets the requirements of the project. During the research and development, depending on the changes or upgrades of the design, the hardware will require making several versions, and each version of hardware will cost a lot of time to complete tests, which seriously affects the progress of the project. Purchasing an existing universal development platform or evaluation board from the market can save a lot of hardware development time, but these platforms are usually bulky and made with unoptimized power consumption. Usually, a development platform only contains one type of processing chips. For examples, Arduino and ARM mbed development boards only use MCU chips. And Raspberry Pi only uses the SoC chip. In a complex project with strict requirements for real-time computing, a single processing chip may restrict the development of programming and function implementation. If a single high-performance microcontroller handles all the tasks, the real-time performance of the calculation becomes worse and worse as the number of sensors and functions increases.

This thesis introduces an FPGA and MCU-based stackable processing platform that provides similar benefits as a customized designed platform without increasing the time of design and test hardware from developers. Based on author's years of work experience in university labs and start-ups, combined with literature reviews, it was concluded that the robots, whether traveling on land, in the water or in the air, all involved applications that were very similar but yet somewhat different. That's when the idea took hold to develop a computational ecosystem that can be easily tailored and adapted to the different application needs of individual developers. This computational ecosystem consists of different modules that allow a platform to contain multiple processors. Multiple communication channels between different processors and sensors are formed with the stackable interface design.

In a project with complex control calculation, developers can classify tasks and operations based on the different requirements of the real-time calculation, and ultimately execute different levels of tasks on different processing modules. The goal is to deliver a performance-optimized stackable hardware platform with a compact size, which has a variety of interfaces and scalable computing power, allowing people to focus on achieving their ideas and saving development time. Developers can mix and match a module's individual capabilities according to the project's needs, making every solution a custom tailored solution without the headache of a custom built platform.

Chapter 2

System Description

2.1 Introduction

In the field of computers, the concept of stackable electronic hardware was initially motivated by the demand for memory size and computing power to obtain larger storage space and better computing performance by integrating the contribution of each device. This thesis presents a stack-able development platform with extensible peripherals. For this platform, the extension of computing power is not the only factor considered when designing the hardware stackability feature also allowing the collaboration between different chips provides distinguishing functionality. This platform offers the possibility of having multiple Field-Programmable Gate Arrays(FPGAs) and Micro-Controller Units(MCUs) work together in harmony. Based on the characteristics of FPGAs and MCUs, this collaboration integrates multiple advantages to fulfill the growing demand for applications, especially for hard real-time processing applications.

In the research and development of electronic robots or mobile devices, choosing to design a customized computing platform on its own or purchase an existing development platform from the market is usually the first problem developers face. Comparing the customized development platform and general development platform purchased from the market, they have their own advantages and disadvantages. The customized development platform can meet the requirements of developers in terms of size, performance and power consumption. However, graduate students and developers need to spend a lot of time and energy to complete the preparatory work for hardware design and create a hardware development platform that meets the requirements of the project. During the research and development, depending on the changes or upgrades of the design, the hardware will require making several versions, and each version of hardware will cost a lot of time to complete tests, which seriously affects the progress of the project. Purchasing an existing universal development platform or evaluation board from the market can save a lot of hardware development time, but these platforms are usually bulky and made with unoptimized power consumption.

Many development platforms only contain one type of processing chips. For examples, Arduino and ARM mbed development boards only use MCU chips. And Raspberry Pi only uses the SoC chip. In a complex project with strict requirements for real-time computing, a single processing chip may restrict the development of programming and function implementation. If a single high-performance microcontroller handles all the tasks, the real-time performance of the calculation becomes worse and worse as the number of sensors and functions increases.

An FPGA and MCU-based stackable development platform is introduced as ioNeurons^{TM}, which also can integrate with an onboard computing module Intel$^{®}$ Edison that can run Linux. This stackable processing platform provides similar benefits as a customized designed platform without increasing the time of design and test hardware from developers. This concept is illustrated in Figure 2.1. FPGAs and MCUs have their own advantages, and the ioNeurons^{TM} platform takes advantage of the FPGA, which provides parallelism and has far more capability under the same power requirements as a microprocessor. Without an Operating System(OS), FPGA circuitry is genuinely a hard implementation of program execution, which is more reliable. On the other hand, MCUs provide better portability of design. Generally, MCUs have shorter development iterations and better floating point operations capabilities. FPGA and MCU cooperation can offer better solutions not only in computing speed but also in power consumption and device size to meet size, weight, and power(SWaP) requirements.

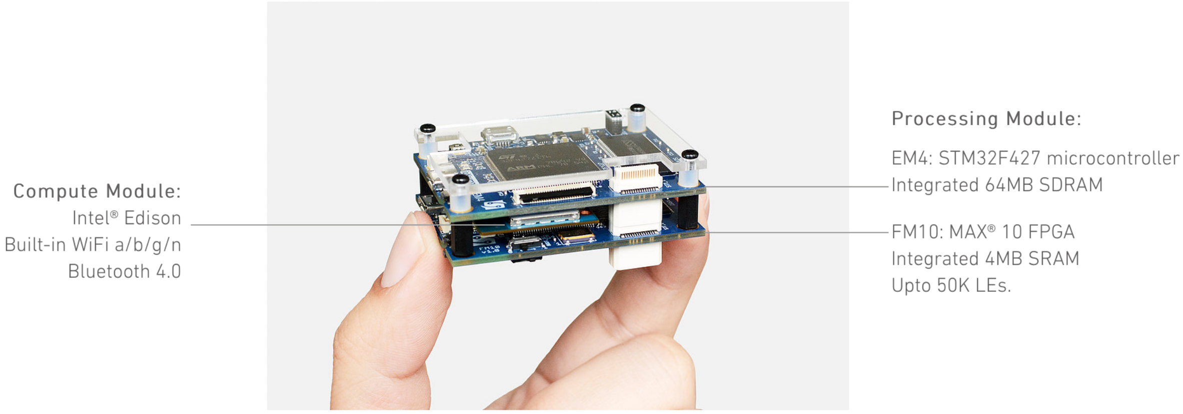


Figure 2.1: ioNeurons$^{\text{TM}}$ platform overview

The ioNeurons$^{\text{TM}}$ platform includes three types of modules, a processing module, a sensing module, and an interface module. Based on a particular interface design of modules, ioNeurons$^{\text{TM}}$ processing modules are stackable in no specific order. The number of modules to be stacked is not limited by the power supply, because a power connector and an individual voltage regulator are embedded on each board. The integrated power management chips guarantee a stable power source to each module and ensure every component is working in well-built condition.

The combination of ioNeurons$^{\text{TM}}$ modules is flexible. A platform could contain one FPGA processing module, and one MCU processing module as shown in Figure 2.1.

However, it may contain two MCU processing modules, and two FPGA processing modules to provide more computing power to developers. This concept is illustrated in Figure 2.2.

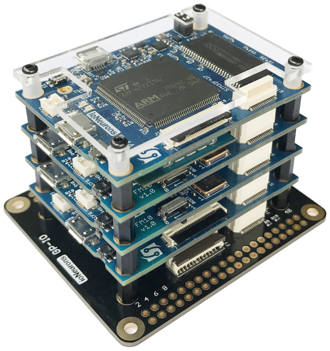


Figure 2.2: ioNeurons^{TM} platform assembled by four processing modules

As of now, ioNeurons^{TM} has two processing modules: FM10 and EM4. FM10 is a Field-programmable gate array(FPGA) processing module built with an Intel® MAX® 10 FPGA. The ioNeurons^{TM} FM10 introduces a compact-sized FPGA development module suited for prototyping circuit designs such as robots and digital signal processing projects. FM10 can work standalone; however, with the design of stackable connectors, it can also integrate with multiple FM10 or EM4 modules.

EM4 is a Micro-Controller Unit(MCU) processing module built with ARM® Cortex®-M4 core STM32 microcontroller in LQFP114 package and an Intel® Edison computing module. The ioNeurons^{TM} EM4 allows users to easily develop embedded systems and Internet of Things(IoT) applications with a STM32F427XX high-performance MCU and the Edison computing module that interfaces with the EM4 via a 70-pin connector. The EM4 module also integrates a 64-Mbit high speed SDRAM.

For the sensing module, ioNeurons^{TM} C5M is a multi-sensors module that provides reliable color image data, motion data, and range data, which is used in the dual-sensor imaging platform as illustrated in Figure 2.3. It does not only include a

5M-pixel Complementary Metal-Oxide-Semiconductor (CMOS) image sensor, it also comes with an Inertial Measurement Unit (IMU), a magnetometer and an optical Time of Flight (ToF) range sensor. The design of the C5M sensing module will be introduced in detail in Chapter 5.

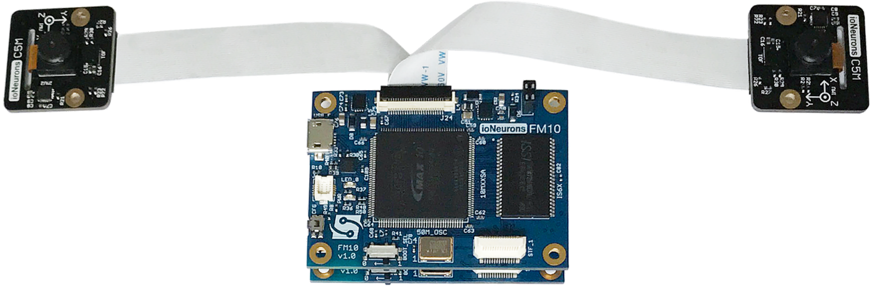


Figure 2.3: FPGA-based dual image sensor development platform

These modules are team players; developers can mix and match a module's individual capabilities according to the project's needs, making every solution a custom tailored solution without the headache of a custom built platform. As shown in Figure 2.3, it is a dual image sensor development platform built with ioNeurons^{TM} modules, which have robust real-time image processing capability. It can be used to implement Simultaneous Localization And Mapping (SLAM) algorithms for robotic mapping and navigation. The platform has two FPGA processing modules and two 5 megapixel color cameras as sensing modules. Each of the cameras is deployed with an independent FPGA and SRAMs to process the raw image data, and the two FPGA modules can also communicate with each other. Comparing to an MCU-based platform, the ioNeurons^{TM} based platform takes advantage of the FPGA which can implement image processing algorithms much faster with parallel computing power. Compared to using a PC graphics card to perform image processing, an ioNeurons^{TM} modules based platform has a smaller volume and a lower power consumption. This dual image sensor development platform can evaluate algorithms fast, and also meet size, weight, and power (SWaP) requirements.

Interface modules can be used as expansion boards allowing ioNeurons^{TM} modules to be applied in different application environments to meet various requirements, or to be used to evaluate ioNeurons^{TM} modules directly. So far, there are two types of interface module: a general purpose IO module and an RPi-IO module. The general purpose IO module is at the bottom (shown in Figure 2.2), which connects to the processing modules and leads the GPIO out to the vias on the PCB so that users can access the processing modules and attach peripherals.

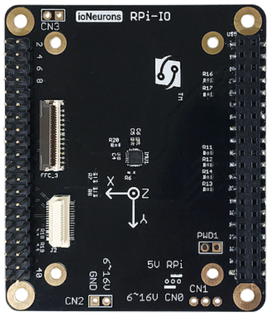


Figure 2.4: RPi-IO interface module

The RPi-IO module allows connections to the processing modules, and it also integrates a 40-pin standard female connector that can connect to a Raspberry Pi2/3. The RPi-IO can be considered an adapter module for connecting a Raspberry Pi and ioNeurons^{TM} FPGA or MCU modules to enhance the computing power of Raspberry Pi. FPGA and MCU modules free up the Linux OS based Raspberry Pi for higher Level tasks, which is especially useful for real-time processing. RPi-IO also offers a motion tracking sensor that combines a 3-axis gyroscope, a 3-axis accelerometer, a 3-axis magnetometer and a Digital Motion Processor in a single chip. Figure 2.5 shows the application platform powered by ioNeurons^{TM} processing modules and Raspberry

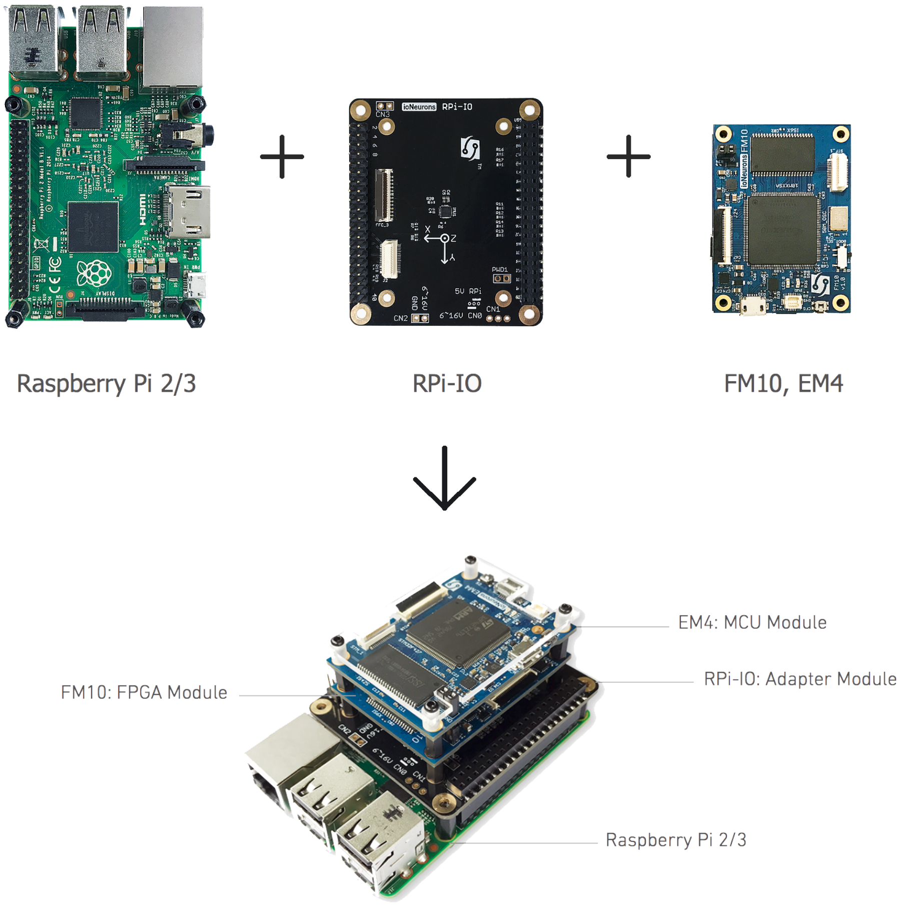


Figure 2.5: ioNeurons$^{\mathrm{TM}}$ modules incorporated with Raspberry Pi

Pi through RPi-IO.

2.2 Platform Architecture

The ioNeurons^{TM} platform consists of three module types. They are categorized into processing modules, sensor modules, and interface modules based on functionality. Figure 2.6 shows the basic ioNeurons^{TM} system block diagram. The processing module is the core part of the whole platform. It can connect to sensor modules and even customized interface modules to meet different application requirements.

Every processing module can be powered individually to avoid the interference of a power source, and the output power limitation of a Power Management Integrated Circuit (PMIC). Every processing module possesses its own stackable interface enabling connections between modules or interface modules that provide connectors to peripherals. If a sensor module needs to pass the collected information to multiple processing modules, this sensor module must establish a connection with those processing modules through the stackable interface which supports multiple communication protocols to establish a point-to-multipoint communication network. If a sensor module only needs to deliver its data to one processing module, connection through the flexible flat cable(FFC) connector would be satisfactory. The FFC connector can output 3.3V to power a sensor module, while it provides 20 customizable pins in an FM10 module, and SPI, a UART interface, GPIOs in an EM4 module.

For the EM4, the Intel® Edison is an additional optional computing module that contains core system processing and connectivity elements, such as a processor, Wi-Fi, and Bluetooth. The Intel® Edison interfaces with EM4 via a 70-pin connector, and EM4 provides input power to the Edison. The Edison module can communicate with a random EM4 and FM10 through the stackable interface in a stacked module

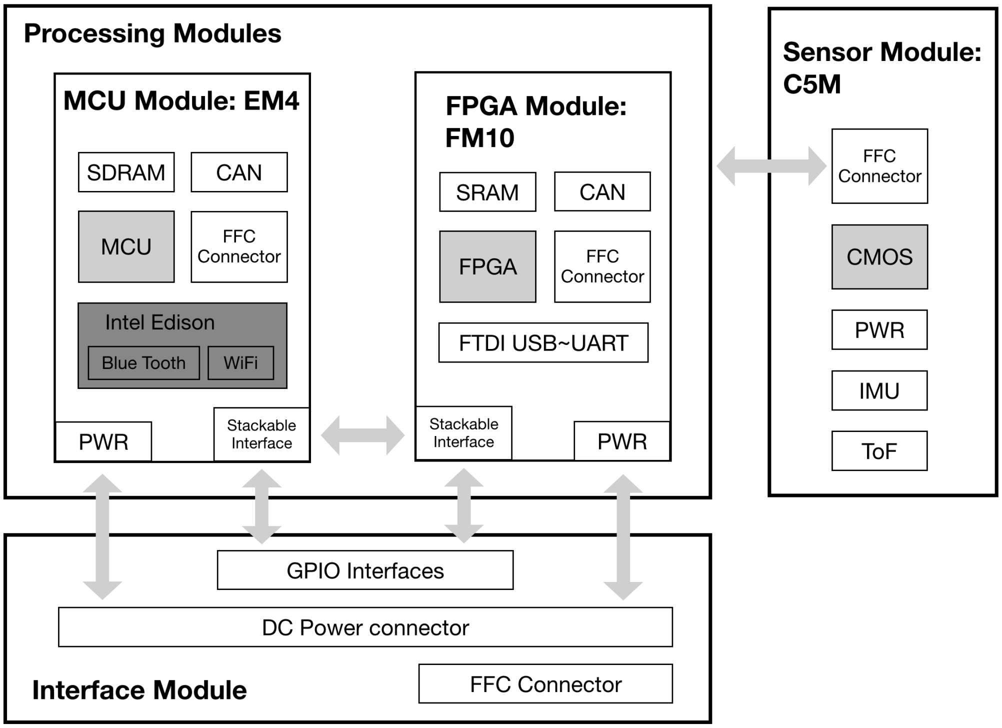


Figure 2.6: ioNeurons$^{\text{TM}}$ system block diagram.

combination. Moreover, two Edison modules can also communicate with each other through the stackable interface if they are in two adjacent EM4 modules.

2.3 Stack-able Interfaces

In modern electronic hardware design, modularity has become a trend that makes coordination between different products easier and more cost-effective. In this case, varied types of board-to-board connectors become essential components to achieve connection between modular hardware, and there are thousands of different connectors that can be selected to meet various requirements in projects.

Board-to-board connectors can be categorized by opening direction, mounting type, transfer speed, and functionality. There are two types of opening direction: right angle to board or parallel to board. In terms of mounting type, they can be separated into two groups: surface mount and through-hole. In general, through-hole mounted connectors are stronger, cheaper, and easier for tracing than surface mounted connectors; however, their size tends to be more prominent when having the same amount of pin positions as surface mount connector because of the wider pitch between every two pins. Also, PCB tracing can be significantly affected in the area where the through-hole-pins present, because tracing cannot be applied in the inner layers of PCB, thus leading to an increase of the overall dimensions of the PCB. Based on data transfer speed, there are high-speed connectors and low-speed connectors. Based on functionality, there are data bus connectors, power connectors and combined connectors. Data bus connectors are designed for data transfer, in which the pins are ordinarily small, while power connectors are designed for power supply, in which the pin size is big enough to carry current safely. Combined connectors have both advantages that data bus connector and power connectors have, which implies more expensive cost.

Considering functions and cost-benefit factor, the stack-able interfaces are implemented with a pair of P5KS series narrow pitch, 30-pin, board-to-board connectors from Panasonic Corporation. The header and receptacle are soldered separately on the top and bottom side of each computing module. Every two modules then can be connected via this type of connector; the connector receptacle at the top of one module connects to the connector header at the bottom of another module, integrating with four plastic spacers of a certain height. Figure 2.7 presents the structure of the stack-able connector.

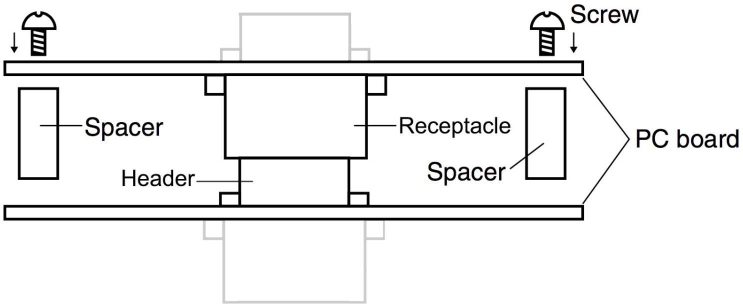


Figure 2.7: ioNeurons$^{\text{TM}}$ stackable connectors

The Manufacturer part number of the header connector is AXK6S30547YG. This mating receptacle connector is available in three different heights. The board-to-board mating stack height can be 5.0, 5.5, or 8.0 millimeter. Table 2.1 lists the Panasonic part number for the receptacle connector and corresponding DigiKey part numbers for an expansion board. The header of the P5KS connector is on the top, while the socket of the connector is on the bottom. The socket and header are protected from reverse mating by two chamfered corners. When the Intel® Edison is not present, mating height 5.0 and 5.5 mm receptacle parts can be used to lower the overall height. Otherwise, when the Intel® Edison is present, a mating height 8.0 mm receptacle

Table 2.1: Stackable 30-pin P5KS connector part numbers

Panasonic P/N	**Mating stack height**	**Digikey P/N**
AXK5S30047YG	5.0 mm	255-2564-1-ND
AXK5S30247YG	5.5 mm	255-3234-1-ND
AXK5S30347YG	8.0 mm	255-2563-1-ND

must be chosen to ensure a large enough space between two processing modules to accommodate the Edison module. Figure 2.1 shows the presence of an Intel® Edison between an EM4 module and FM10 module with a pair of 8.0 mm mating height connectors.

When designing the stackable interface, a thorough consideration of the flexibility of the connector function is vitally important. Each processing module must be able to satisfy the requirement of high-speed communication between nodes performing distributed collaborative computing. The stackable interface can establish connections for any two processing modules in the stack structure; it also provides the dedicated communication channels supporting multiple protocols for any two adjacent modules. The stackable interface also offers high-speed communication connections between modules.

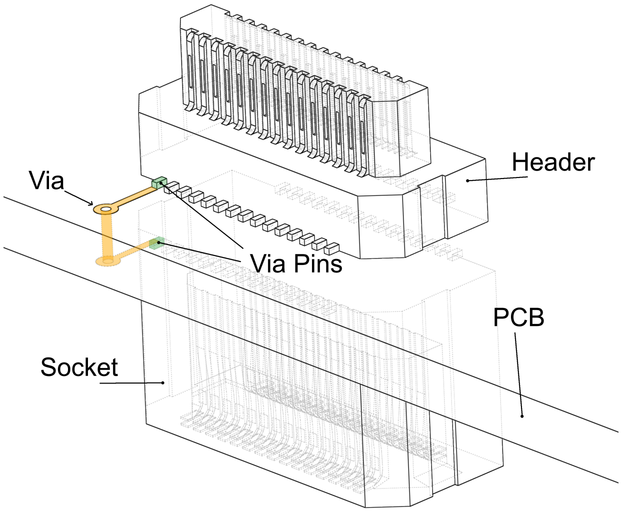


Figure 2.8: Via Pins in the stackable connector

Based on the connecting type, the 60 pins on a pair of the connectors can be divided into two categories: Independent Pin and Via Pin. In this thesis, it is defined that an independent pin is not connected to any other pin of the connector, but directly connected to a pin of the processor or other chips. The Via Pin is defined as any two pins which have a corresponding position on the top header and bottom socket respectively, and are connected through a via in the PCB. Via Pins exist in pairs carrying the same signal. Via Pins provide powerful support for device expansion and cross-module communications. Figure 2.8 shows a pair of Via Pins on the stackable interface.

Via Pins can be categorized as Power Via Pins, Databus Via Pins and Data Pass Via Pins. The definitions of each type are as follows:

Power Via Pins connect a common power source to interface modules or sensor modules and provide a common ground for all of them.

Databus Via Pins connect to a specific pin on the processor of the module. Databus Via Pins not only provide a channel for data exchange, but also allow the processor to act as a node to connect to the data exchange channel. For example, a pair of Databus Via Pins having corresponding positions on multiple processing modules can be connected together to form a channel in the stack structure. This channel passes through all the modules enabling data exchange between any two processors. In I2C communication protocol, these processor nodes are connected to SDA and SCL formed by the Databus Via Pins.

Data Pass Via Pins do not connect to any pins on the processor or any other signals on the module. This type of via pin is isolated from the surroundings, which is to provide a channel allowing the upper layer module to communicate with the lower layer module in a multi-layer processing module stack structure.

Figure 2.9 shows the different types of stackable pins on three processing modules. There are 4 pairs of Via Pins that are set as Power Via Pins: 2 for VCC and 2 for GND. This reinforced dual pair of Via Pins design effectively avoids the possibility of burning the connector due to overload current on a single pair of pins, thereby increasing the capacity to take and supply large current flow.

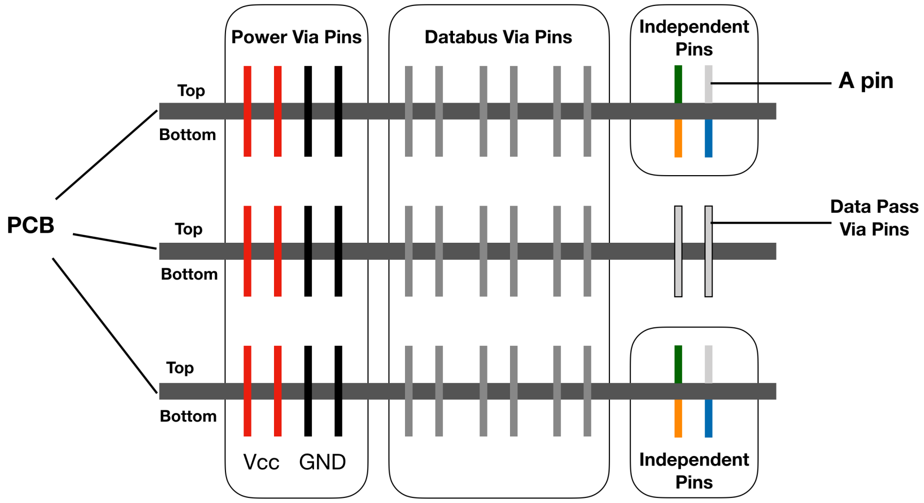


Figure 2.9: Types of Via Pins in the stackable connectors

Chapter 3

Design and Implementation of the FPGA Module

3.1 FM10 Board Overview

The FM10 is a compact-sized FPGA processing module in the ioNeurons^{TM} series. It is capable of performing Hard-Real-Time digital logic operations in parallel; It also contains huge resources of Logic Gates(LGs), an 8-channel Analog-to-Digital Converter (ADC), and a 4MB Static Random Access Memory (SRAM) to implement complex digital computations. The primary function of the ADC is to translate analog quantities to digital data for further data processing. The SRAM provides users high-speed access memory with fully static operations, no clock or refresh required. FM10 can take a wide range of voltage input from 3.3V to 16.0V benefiting from an efficient battery management chip; Meanwhile it can provide a 3.3V DC power source at maximum 1A current.

The physical dimension of FM10 is 40.0 mm x 55.0 mm. It has one 24-pin Flat Flex Cable (FFC) connector and two 30-pin P5K series connectors, which permit 40 GPIOs in total and outputs of 3.3V or 5.0V voltage individually. FM10 can easily transfer data to computers because it integrates a Future Technology Devices International (FTDI) USB-to-Serial port chip and a USB type-AB connector.

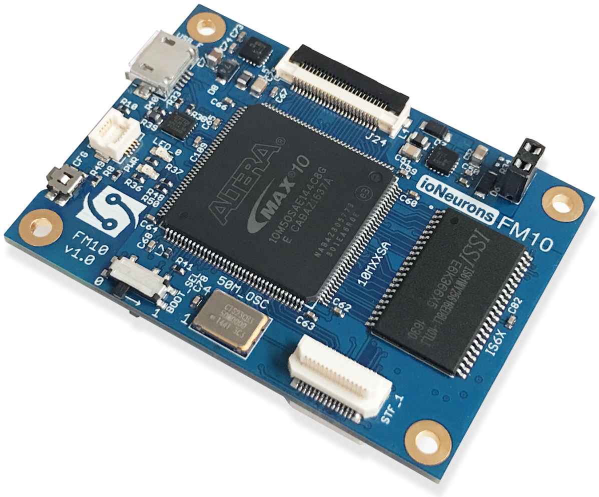


Figure 3.1: ioNeurons$^{\text{TM}}$ FM10

The FM10 can work standalone; however, with the design of the stackable connector, it can also collaborate with multiple FM10 or EM4 modules, which can enhance the computing capacity and increase the number of Input/Output interfaces. FM10 is equipped with user programmable LEDs, push buttons and slide buttons, which allows developers to apply different configurations to meet specific requirements.

The FM10 uses an Intel Max 10 series FPGA chip 10M25SAE114I7G (144-pin EQFP); a single-chip, low-cost, programmable logic device with 25000 Logic Elements(LEs). It also integrates a non-volatile user flash memory and ADCs in a hard IP core block. The ADCs allow the FPGA to process analog signals from real-world sensors. It provides 8-channel 12-bit digital representation of the analog signal being observed with a cumulative sampling rate of up to 1 million samples per seconds (MSPS).

3.2 Layout and Components

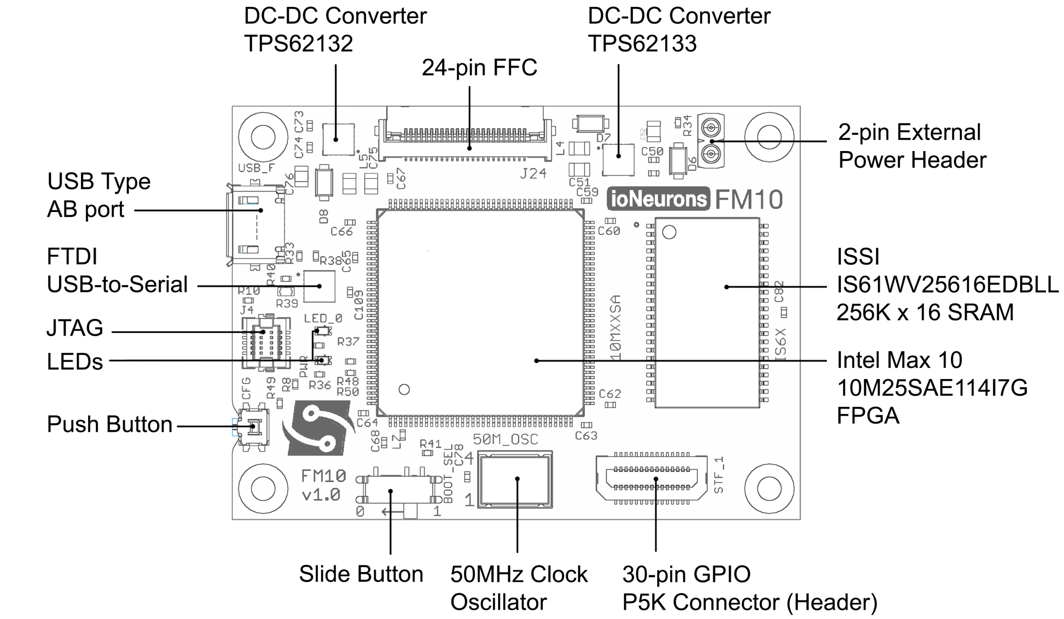


Figure 3.2: FM10 module PCB and components (top view)

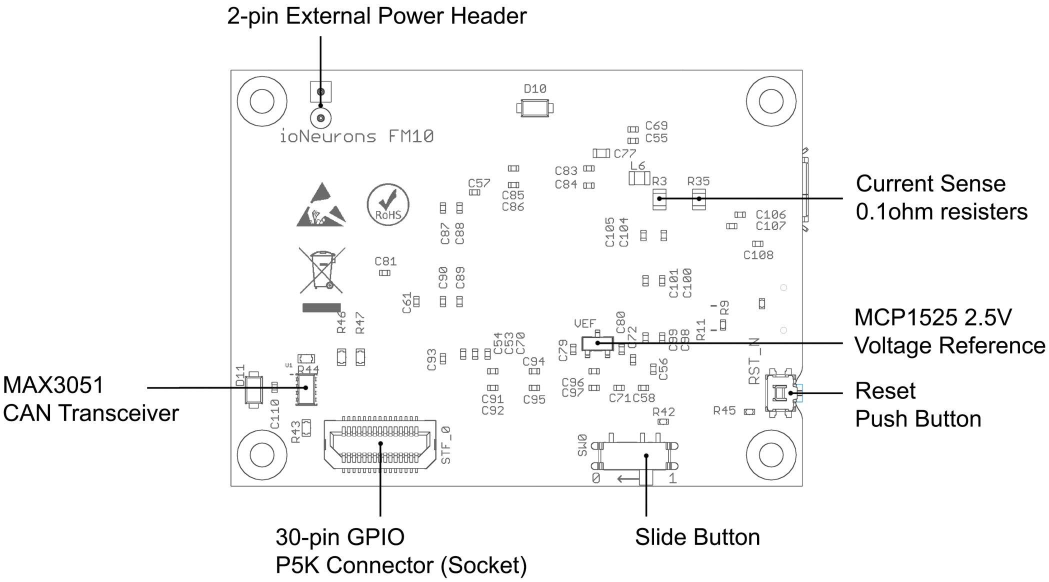


Figure 3.3: FM10 module PCB and components (bottom view)

It is vitally important that every component is placed in a suitable position on a printed circuit board since it determines the overall dimension, number of layers, and complexity of tracing. Taking production cost and manufacturing techniques into account while keeping the overall size as small as possible, the FM10 employs a 6 layer PCB structure and plating through holes only; no blind via holes or buried via holes were used since they significantly increase the cost of production. The minimum diameter of through holes is 0.15 mm. The solder resist layer covers any through holes which are under 0.35 mm in diameter to protect them and avoid a short circuit, and the stop layer masks will not be applied to these hole locations. The minimum width of a trace is 0.1524 mm, and the minimum clearance between different signal wires is 0.1778 mm. The minimum distance between objects on signal layers and the board dimension is 0.2032 mm, and the thermal isolation is 0.1778 mm. The FR4 insulation material is applied between any two conductor layers. Table 3.1 shows the PCB layer stack. Electroless Nickel Immersion Gold (ENIG) surface treatment is applied on the top and bottom layer during PCB manufacture to prevent the nickel from oxidizing with a thin layer of immersion gold.

When determining the relative position of components, the one having the most pins takes priority because its position has the most significant impact on tracing. The FPGA chip (10M23SAE114I7G) has 144 pins; it is the chip with the largest number of pins on this board. If one side of this chip, where pin number 1 is located, is defined as the chip's left side, most of the high-speed pin and GPIO of the FPGA are located on the bottom side (pin37 to pin72) and right side (pin73 to pin108). Relatively, the SRAM and the FFC connector are the two components that have the next most pins; to reduce the distance between them and the FPGA, the SRAM is placed below the FPGA chip while the FFC connector is placed on the right of the FPGA. In addition, the 50 MHz crystal is placed as close to the FPGA as possible because of its high-frequency clock signal, which is a critical factor to maintain stable

Table 3.1: FM10 PCB Layer Stack

Layer	**Layer Stack**	**Layer Name**	**Material**	**Thickness**	**Plane**
1		Top paster	Sn63/Pb37	0.12mm	
2		Top overlay			
3		Top solder	Solder resist	0.01mm	
4	- - - - - - - - - - - - -	Top ENIG	Gold	0.05μm	
5	— — — — — —	Top	Copper	0.018mm	Signal
6	///////////////	Dielectric 1	FR-4 TG140	0.0913mm	
7	— — — — — —	Layer 2	Copper	0.03mm	D_GND
8	□ □ □ □	Dielectric 2	FR-4 TG140	0.53mm	
9	— — — — — —	Layer 3	Copper	0.03mm	Signal
10	///////////////	Dielectric 3	FR-4 TG140	0.1056mm	
11	— — — — — —	Layer 4	Copper	0.03mm	5V_SYS
12	□ □ □ □	Dielectric 4	FR-4 TG140	0.53mm	
13	— — — — — —	Layer 5	Copper	0.03mm	A_GND
14	///////////////	Dielectric 6	FR-4 TG140	0.0913mm	
15	— — — — — —	Bottom	Copper	0.018mm	Signal
16	- - - - - - - - - - - - -	Bottom ENIG	Gold	0.05μm	
17		Bottom solder	Solder resist	0.01mm	
18		Bottom overlay			
19		Bottom paster	Sn63/Pb37	0.12mm	

operation of the whole system.

In order to reduce electromagnetic interference (EMI), the FM10 6-layer structure adopts the top layer, the third layer, and the bottom layer as signal layers, in which copper is paved in the area around signal routes and connects to the ground plane. The second and the fifth layer are the ground planes, and the fourth layer is the power plane. The solid ground plane will minimize inductance, and any signal plane close to the ground plane will have continuous return current.

Decoupling capacitors are placed near the power pins of each chip. When the chip's payload suddenly increases, the circuit tries to increase its current, however the inductance in the power supply line hinders the current rising and causes the power line voltage to sag. At this time the decoupling capacitors provide short bursts of current to maintain the voltage at a stable level.

3.2.1 Power management and distribution

The FM10 can be powered by a DC power supply, USB, and Li-Ion or other batteries because of its high-performance power management module design which has a wide operating input voltage range from 3.3 V to 17 V. It allows FM10 to easily work in various application environments with flexible power requirements. At the same time, FM10 can provide 5 V and 3.3 V power sources, which allows small sensor modules that connect to FM10 to be powered, thus avoiding redundant power circuit design for external sensors.

The power management module of FM10 has two synchronous Step-Down DC to DC Converters from Texas Instruments; their part numbers are TPS62133 and TPS62132. Both of them adopt 16-pin 3x3 mm QFN packages and can provide 3 A of continuous output current at 3.3 V and 5 V respectively with a maximum up to 95% converting efficiency. These two chips are made for mobile application platforms;

They use advanced regulation topology (DCS-Control) that combines advantages of multiple control modes, which can monitor output voltage changes and feed it directly to a fast comparator to set the switching frequency. This technology provides immediate response for dynamic load changes to achieve fast and stable voltage operation. These DC to DC converters also prevent the circuit from shorting and overloading. When a load or a short circuit is detected and the output voltage drops below 0.5V, the current limit will be reduced to 1.6A immediately. If the output voltage picks up above 0.5V again, the device recovers to the normal operating mode. Figure 3.4 shows the schematic of the voltage converter connectivity in FM10. A 1.0uH inductor and a 22.0uF capacitor are connected to its output as an external LC output filter.

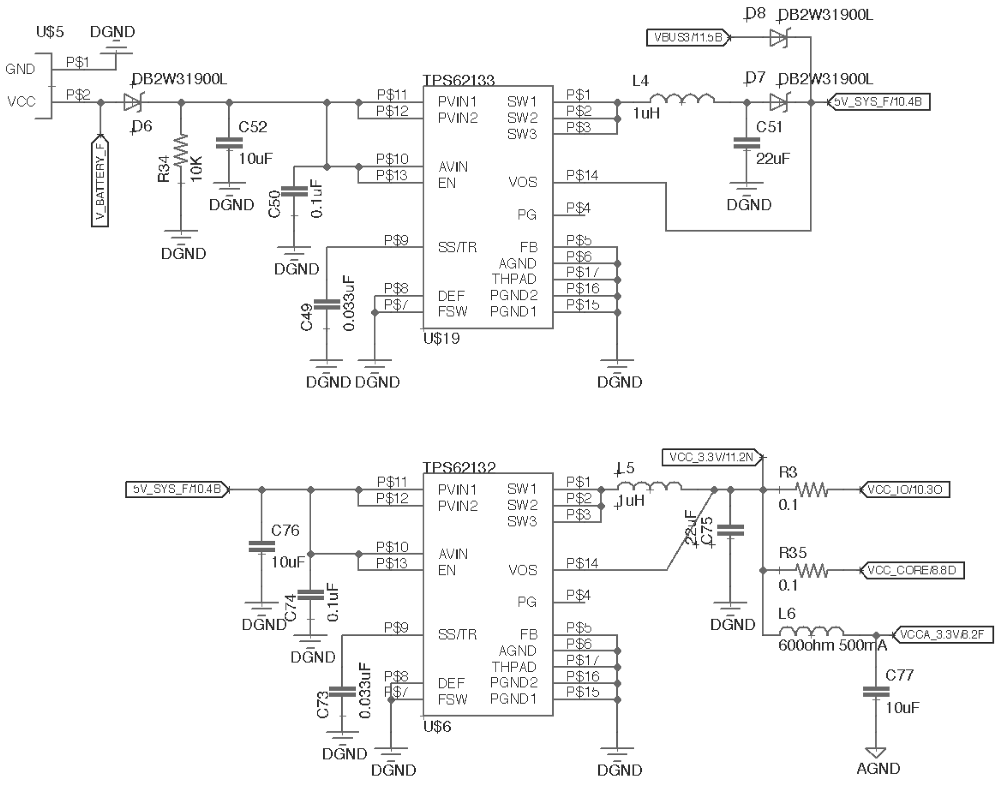


Figure 3.4: FM10 module power schematic

The chip, TPS62133, accepts a maximum of 17.0 V input power, and its output voltage is fixed 5.0 V, which is the input voltage for the next level DC-DC converter chip TPS62132. A DB2W900L diode from Panasonic Electronic Components is placed between the two chips to prevent reverse current. Moreover, the USB connector also has a 5.0V power line connecting to the input of TPS62132 via a DB2W900L diode, and this allows the system to also be powered by a USB connector directly. TPS62132 provides 3.3 V output voltage to the FPGA and other chips whose rated voltages are 3.3 V. The two diodes (D8, D7) feature 30.0 V reverse breakdown voltage which prevents the USB from being burnt by a high reverse voltage, and allows FM10 to be powered by battery and USB at the same time.

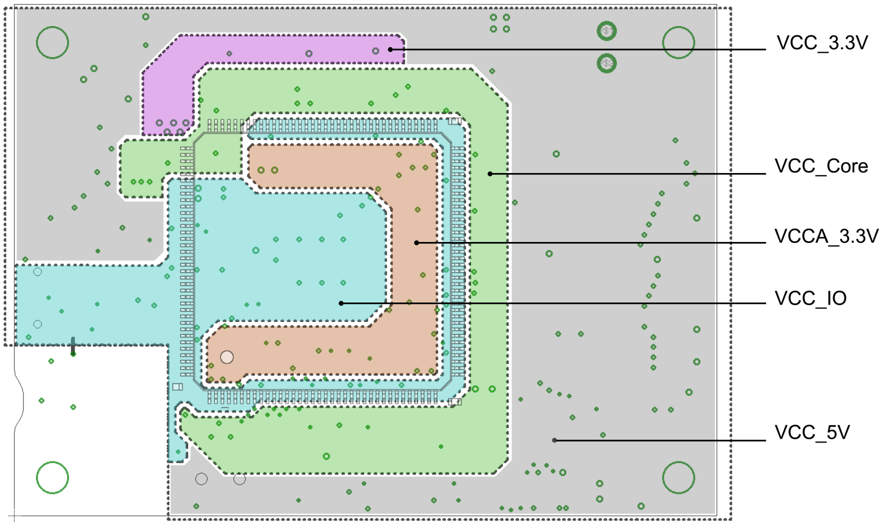


Figure 3.5: Power layer design of FM10's PCB

In order to provide the FPGA chip with a clean, low-noise power supply, a power tree was designed to separate the 3.3V external power supply into different groups serving as multiple inputs to the MAX 10 device via VCC_ONE, VCC_IO and VCCA. Figure 3.5 shows the power layer design in the PCB. The VCC_3.3V power supply is divided into VCC_Core, VCC_3.3V and VCC_IO to power different portions varying in function on the FPGA chip. This reduces the interference between different functional groups which can arise due to voltage fluctuations caused by the different current demands of each functional group.

3.2.2 FPGA: 10M25SAE114I7G

The chip, 10M255AE144I7G, belongs to Intel® MAX10 series; it is single-supply FPGA, which requires only 1 external power supply of 3.3 V [14]. It is built on 55nm embedded NOR flash technology, providing 25000 logic elements, integrated Analog-to-Digital Converters (ADC) and 2 MB of user flash memory to store configuration files and user data. The ADC provides 8 channels with 12-bit digital representation of the analog signals being observed, with a cumulative sampling rate of up to 1million samples per second (MSPS). The user flash memory provides ideal storage space to store non-volatile information. It uses an EQFP package, offering 144 pins, in which 101 pins have general-purpose input/output (GPIO) functionality. In addition, the speed grade -7I is another important parameter in design. There is no consistent definition of a speed grade for all FPGA/CPLD devices. The -7I for this chip means it is industrial class; the pin-to-pin delay is 7 nanoseconds. In other words, the smaller the number, the faster the speed, which has an impact on the selection of SRAM.

The maximum power consumption of the 10M25S chip can be 2.943 W, which comes from the VCC_ONE pins. All VCC_ONE pins are powered by VCC_Core. If the user's design is beyond the maximum power consumption, it will cause that part of the function to perform abnormally. The minimum system design includes the power supply, crystal oscillator, voltage reference and adapter interface. The type of crystal oscillator is CB3LV-3C-50M0000 from CTS Electronics Components providing a 50 MHz clock signal. The voltage reference chip, MCP1525 from Microchip, provides a 2.5 V precision voltage reference that has initial maximum tolerance of ±1%, and temperature stability of ±50 ppm/c^o.

3.2.3 SRAM: IS61WV25616EDBLL

Static Random Access Memory (SRAM) is a kind of random access memory. The 'static' means the memory can store the data permanently as long as it is powered. Data does not need to be refreshed periodically and would not be lost. By contrast, the data in SDRAM needs to refreshed periodically to ensure the effectiveness of the data. Both of them are called volatile memory since the stored data would be lost if the power is cut off. Compared to SDRAM, the SRAM has advantages of faster speed, lower power consumption, especially during idle, and easier control. This is also the reason that SRAM is selected for FM10. The disadvantages of SRAM are more complicated internal circuit and high production cost, therefore, it is not suitable to be used in the applications requiring large storage.

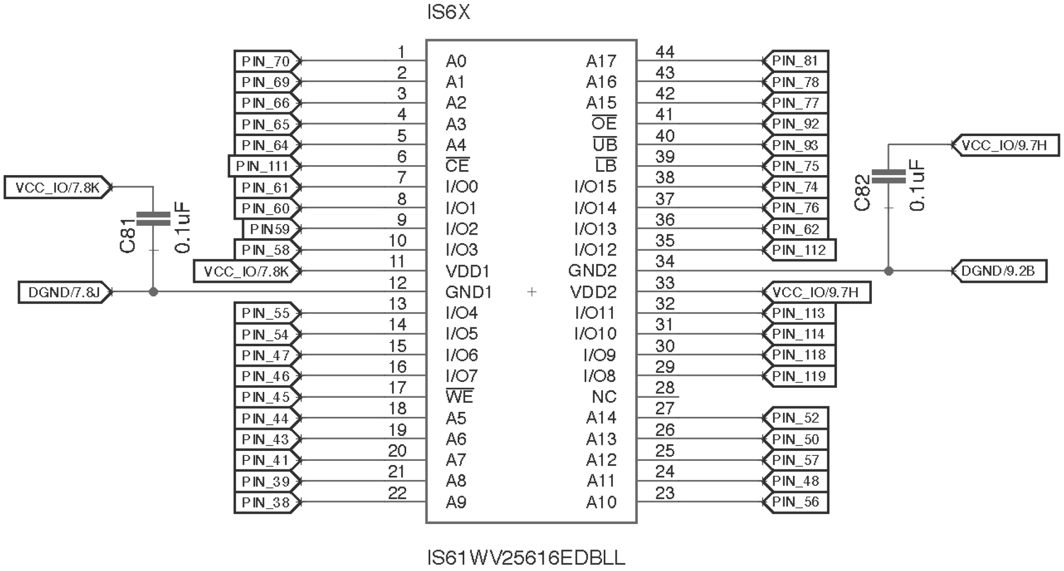


Figure 3.6: The SRAM schematic of FM10 module

The ISSI, IS61WV25616EDBLL, on FM10 is a high-speed asynchronous CMOS SRAM chip, which has 4194304 bits of memory organized as 262144 words by 16 bits. This chip can satisfy most of the application requirements that need high computing speed. The asynchronous means that the read and write operation is triggered by rising or falling edge of the control signal or the address signal. The read and write can occur at any time. However, synchronous SRAM has one clock signal, thus all the operations are driven by the clock signal. In other words, the read and write operation only happens at expected times. The access time of IS61Wv25616EDBLL is 8 ∼ 10 ns, which is faster than the FPGA's clock signal. In another words, the SRAM can be accessed and operated by the FPGA on every clock cycle meaning there are no waiting clock cycles. Figure 3.6 shows how the SRAM is powered in FM10 and its pin connection to FPGA.

3.2.4 CAN Transceiver

A CAN bus standing for Controller Area Network bus is a robust data communication protocol that was initially used on vehicles enabling devices to communicate with each other, but now it is not limited to use in the automobile field only. CAN's high performance and reliability have been recognized, and are widely used in industrial automation, ships, medical equipment, and so on. The CAN transceiver makes the FM10 interfaces more friendly to access CAN network, and this greatly increases the applications of FM10. The CAN bus provides powerful support for distributed control platform which is using ioNeurons$^{\text{TM}}$ processing modules, to realize real-time and reliable data communication between various nodes.

Unlike other data transfer protocols, a pair of CAN bus twisted wires allows more than two devices to connect in parallel and exchange data; in other words, it is not a fixed point-to-point communication protocol; it can create a communication network and offer flexible point-to-point or point-to-multipoint communications. Those devices connecting to a CAN bus are Electronic Control Units (ECUs) also known as nodes. A node can be a simple I/O device or a complicated embedded computer. Any of the devices can be a host as long as it has the corresponding functionality. Once a device broadcasts a message on the bus, every other device will listen and identify if they are interested in this message, and then the message will be received by the one that is interested.

To transfer data, CAN bus signalling uses differential signals on the two wires: one is called CAN_high, the other one is called CAN_low. When CAN_high is driven to 5 V and CAN_low is driven to 0V, the transmission is recognized as dominant state representing logic level 0. When neither of them is driven, they are levelled to the reference voltage which is 2.5 V, and at this time, the transmission is recognized as recessive state representing logic level 1. In the physical layer, these two wires are twisted together in order to reduce the influence of electromagnetic interference. Two 120 Ω termination resistors are placed on each end of the CAN bus to suppress signal reflections.

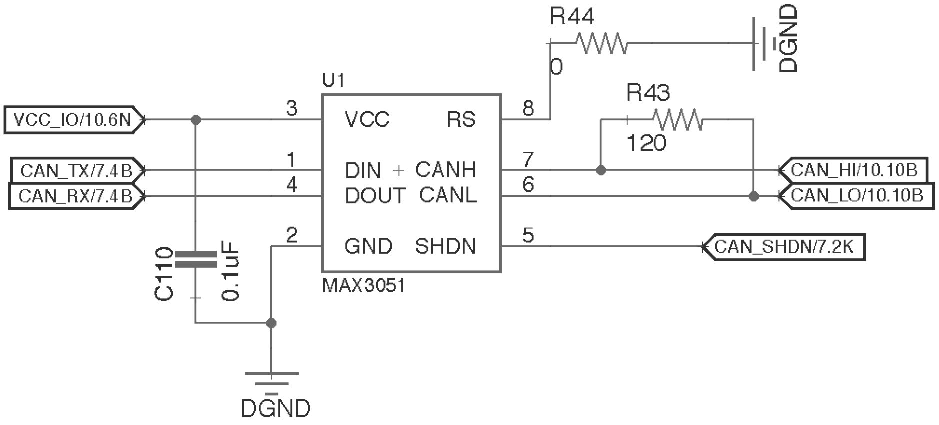


Figure 3.7: The CAN transistor schematic of FM10 module

In order to generate the differential signals, a transceiver chip that is based on CAN bus protocol is required. In FM10, the transceiver chip is MAX3051 from Maxim Integrated. This chip can translate the logic level signal 0 or 1 from the host controller to differential signals that are transmitted on the CAN bus. It also features four different modes of operation that enables data transfer speed control.

The schematic in Figure 3.7 shows a typical operating circuit.

3.2.5 FTDI

As the rapid development of laptops continues toward lighter and thinner, many traditional connectors are replaced by those having lower volume and faster transfer speeds. Universal Serial Bus (USB) facilitates the communication between computer and other peripherals to a large extent. USB is standardized as an industrial standard that prescribes the communication protocol, cable, and connector type. It substitutes the traditional serial communication ports and the connectors that were previously used such as the D-subminiature connector. However, in the development of FPGA and embedded systems, traditional serial communication protocols take less resources, and are relatively easy to implement, so they still play an important role in data transfer and debugging of embedded developments. Universal asynchronous receiver-transmitter (UART) is a very common communication protocol. In order to bridge the connection between USB on the computer and serial output on the FPGA via the UART communication protocol, a UART to USB converter chip from FTDI is needed. A corresponding driver must be installed in the computer operating system, to allow the USB port to be recognized as a virtual serial communication port.

FT234XD USB to basic UART IC provides USB and an asynchronous serial data transfer interface, which is fully compliant with the USB 2.0 specification. Its UART interface has support for 7 and 8 data bits with 1 or 2 stop bits. The data transfer rate can be configured from 300 baud to 3M baud rate. Figure 3.8 illustrates the FT234XD typical configuration with a USB bus powered design. The UART_TXD and UART_RXD are connected to PIN_124 and PIN_120 on the FPGA through 0 Ω resistors or a suitable ferrite bead according to the current; These can reduce the EMI noise that the FT234XD radiates to the USB host over the USB cable.

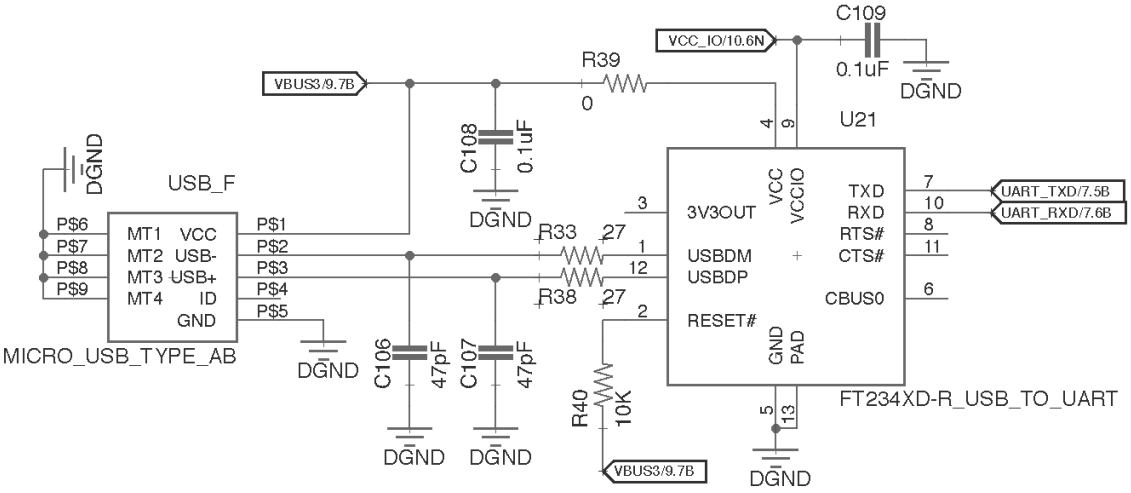


Figure 3.8: The FTDI schematic of FM10 module

3.3 Interfaces and pin mapping

As a processing module, the selection of suitable connectors for FM10 directly impacts the user experience and the way its used. In order to facilitate adoption of FM10 in as many projects as possible and considering the cost-benefit factor, FM10 is designed to use two different types of connector: Flat Flex Cable (FFC) connectors and P5KS series board to board narrow pitch connectors; Both of them are classified as surface mount type and high-speed connectors. The primary function of FFC is to provide and expand the connectivity to other peripherals. P5KS connectors are utilized for implementing expandable and stackable features between ioNeurons$^{\text{TM}}$ modules.

The FFC connector mounted on FM10 module is 24-pin with a 0.4mm pitch, fitting a 12mm wide flat flex cable, which can connect FM10 with another module. In the 24-pin FFC connector, 20 out of 24 pins are defined as data transfer pins, and other 4 pins are set as power pins: 2 for VCC and 2 for GND. This reinforced dual power pin design effectively avoids the possibility of burning the connector due to overloading current on a single pin, thereby increasing the capacity to take and supply large current flow. In addition, the FFC connector has a double-bottom-contact terminal design which can remove dust and contaminants better than single-bottom-contact terminals. Two contact points on each pin also provides stable redundant contact forces for secure electrical reliability. The FFC is placed at upper side of the FPGA chip, and it is very close to FPGA Pin_73 ~ Pin_108. Most of these pins are able to carry high-speed differential signals. Not only does this layout design shorten the trace length between FPGA and FCC connector, but it also reduces the EMI noise. FM10 chooses to use the parallel-to-board type FFC connector to achieve more space between two adjacent boards thus making it easier for users to install cables.

FM10 uses Panasonic P5KS series narrow pitch connectors for communicating between ioNeurons^{TM} processing modules. A pair of 30-pin connectors are soldered on the top and bottom side of FM10. The header of P5KS connector is on the top, while the socket of the connector is on the bottom. The socket and header are protected from reverse mating by two chamfered corners.

Each connector has a total of 30 pins, which are distributed in two rows with a pin-to-pin pitch of 0.5 mm. It was chosen to be used on FM10 due to the diverse mating height choices for this type of connector. In the design, an 8.0 mm mating height is used to avoid collision between the Intel® Edison module and FM10 when the modules are assembled. Both the header and socket have 4 pins that were defined as power pins: 2 for VCC and 2 for GND; each pin can take up to 0.5A current. This reinforced design effectively avoids the possibility of burning the connector due to large current flow, thereby increasing the capacity to take and supply large current flow. A DB2W900L diode from Panasonic Electronic Components is placed between the power source and power pins to prevent current from going backward. In other words, users cannot power FM10 via power pins from the P5KS connector, but FM10 can provide a 5.0 V power output. Table 3.3 and Table 3.4 show the connection

Table 3.2: FM10 24-pin FFC pin-out and signal list

FFC Pin Number	**FFC Pin Name**	**FPGA Pin Name**	**Function**
1	FFC_0	PIN_79	GPIO
2	FFC_1	PIN_80	GPIO
3	FFC_2	PIN_84	GPIO
4	FFC_3	PIN_85	GPIO
5	FFC_4	PIN_86	GPIO
6	FFC_5	PIN_87	GPIO
7	FFC_6	PIN_88	GPIO
8	FFC_7	PIN_89	GPIO
9	FFC_8	PIN_90	GPIO
10	FFC_9	PIN_91	GPIO
11	FFC_VCC	N/A	VCC 3.3V
12	FFC_VCC	N/A	VCC 3.3V
13	FFC_GND	N/A	GND
14	FFC_GND	N/A	GND
15	FFC_10	PIN_96	GPIO
16	FFC_11	PIN_97	GPIO
17	FFC_12	PIN_98	GPIO
18	FFC_13	PIN_99	GPIO
19	FFC_14	PIN_100	GPIO
20	FFC_15	PIN_101	GPIO
21	FFC_16	PIN_102	GPIO
22	FFC_17	PIN_105	GPIO
23	FFC_18	PIN_106	GPIO
24	FFC_19	PIN_110	GPIO

between the 30-pin connectors and the FPGA with pin's functionalities.

In this thesis, it is defined that any pins on the top header and the bottom socket that have the corresponding position to each other and are connected through PCB via, are Via Pins. A pair of Via Pins carry the same signal. The FM10's P5KS connectors contain 16 pairs of Via Pins: some of them are connected to the FPGA's specific pins to build up the I2C or CAN bus communication circuit; while some of them are just pierced forming a bridge without connecting to FPGA. Via Pins can solve issues such as device expansion, module perpendicular connectivity and cross-module communications.

Table 3.3: FM10 30-pin P5KS Header pin-out and signal list

P5K Pin Number	**P5K Pin Name**	**FPGA Pin Name**	**Via Pin**	**Function**
1	HS_1	N/A	Yes	VCC 5V
2	HS_2	N/A	Yes	CAN_LO
3	HS_3	N/A	Yes	VCC 5V
4	HS_4	N/A	Yes	CAN_HI
5	HS_5	N/A	Yes	GND
6	HS_6	PIN_87	Yes	RESET_N
7	HS_7	N/A	Yes	GND
8	HS_8	PIN_122	Yes	DEV_OE
9	HS_9	PIN_135	Yes	GPIO
10	reserved	N/A	N/A	N/A
11	HS_11	PIN_140	Yes	GPIO
12	reserved	N/A	N/A	N/A
13	HS_13	N/A	Yes	N/A
14	H_14	PIN_130	No	GPIO
15	HS_15	N/A	Yes	N/A
16	H_16	PIN_134	No	GPIO
17	HS_17	PIN_25	Yes	GPIO
18	H_18	PIN_26	No	GPIO
19	HS_19	PIN_24	Yes	GPIO
20	H_20	PIN_22	No	GPIO
21	reserved	N/A	N/A	N/A
22	reserved	N/A	N/A	N/A
23	HS_23	N/A	Yes	N/A
24	H_24	PIN_33	No	GPIO
25	HS_25	N/A	Yes	N/A
26	H_26	PIN_32	No	GPIO
27	H_27	PIN_21	No	GPIO
28	H_28	PIN_30	No	GPIO
29	H_29	PIN_17	No	GPIO
30	H_30	PIN_29	No	GPIO

Table 3.4: FM10 30-pin P5KS Socket pin-out and signal list

P5K Pin Number	**P5K Pin Name**	**FPGA Pin Name**	**Via Pin**	**Function**
1	HS_1	N/A	Yes	VCC 5V
2	HS_2	N/A	Yes	CAN_LO
3	HS_3	N/A	Yes	VCC 5V
4	HS_4	N/A	Yes	CAN_HI
5	HS_5	N/A	Yes	GND
6	HS_6	PIN_87	Yes	RESET_N
7	HS_7	N/A	Yes	GND
8	HS_8	PIN_122	Yes	DEV_OE
9	HS_9	PIN_135	Yes	GPIO
10	reserved	N/A	N/A	N/A
11	HS_11	PIN_140	Yes	GPIO
12	reserved	N/A	N/A	N/A
13	HS_13	N/A	Yes	N/A
14	S_14	PIN_123	No	GPIO
15	HS_15	N/A	Yes	N/A
16	S_16	PIN_141	No	GPIO
17	HS_17	PIN_25	Yes	GPIO
18	S_18	PIN_8	No	GPIO
19	HS_19	PIN_24	Yes	GPIO
20	S_20	PIN_10	No	GPIO
21	reserved	N/A	N/A	N/A
22	reserved	N/A	N/A	N/A
23	HS_23	N/A	Yes	N/A
24	S_24	PIN_14	No	GPIO
25	HS_25	N/A	Yes	N/A
26	S_26	PIN_11	No	GPIO
27	S_27	PIN_6	No	GPIO
28	S_28	PIN_13	No	GPIO
29	S_29	PIN_7	No	GPIO
30	S_30	PIN_12	No	GPIO

3.4 FM10 Firmware

This chapter introduces how the FPGA controls and cooperates with the other on-board chips. The SRAM controller and UART transceiver were implemented by the FPGA in Very High-Speed Integrated Circuit Hardware Description Language (VHDL).

3.4.1 SRAM Controller

The SRAM controller can facilitate user access to the data stored in SRAM, control the mode of SRAM read and SRAM write, and change the operating status of the SRAM. In this thesis, the SRAM controller is designed to accomplish the write and read operation of random addresses for 8-bit data through controlling a IS61WV16EDBLL SRAM chip.

In typical FPGA image processing applications, SRAM is used to store image pixel data whose value has a range of 0 ~ 255, each pixel data only needs 8-bits of storage space. ISSI's IS61WV6EDBLL SRAM has 262144 addresses, and each of them can store a 16-bit value. If each 16-bit location is used to store 8-bit data, the maximum occupancy ratio would be 50% meaning a large waste of SRAM storage space. To improve the efficiency of occupancy, the SRAM controller implemented in this thesis can access the upper byte and lower byte of a 16-bit data field separately via configuring the signals on the SRAM's Upper Byte (UB) pin and Lower Byte(LB) pin. When the UB pin is asserted HIGH, and the LB pin is asserted LOW, the write and read operation is only valid for the LB of the data. When the UB pin is asserted LOW, and the LB pin is asserted HIGH, the write and read operation is only valid for the UB of the data. This method gives users 524288 addresses, which is double the number of 16-bit data addresses, with individual 8-bit data storage space. Figure

3.9 illustrates the SRAM controller block diagram with its interfaces.

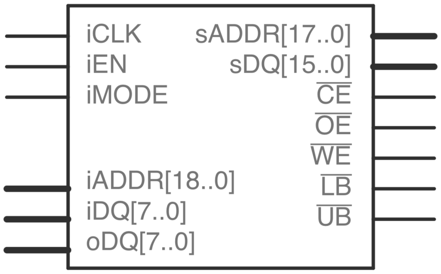


Figure 3.9: SRAM Controller block diagram

The data interface of the SRAM controller is 8-bit wide, and the Address Input of the SRAM controller is 19-bits wide; it consists of the IS61WV6EDBLL SRAM's 18-bit Address Input and 1-bit Byte Control. The Address Inputs of the SRAM from A17 to A0 correspond to the Address Input of the SRAM Controller from A18 to A1. The least significant bit (A0) of the SRAM Controller Address Input is a 1-bit Byte Control which controls the signals on the UB pin and the LB pin. This 1-bit Byte Control allows the separation of a 16-bit storage location into two 8-bit storage locations assigned with individual addresses. Table 3.5 shows the truth table of the SRAM Controller Address Input A0 and UB, LB signals.

Table 3.5: Truth table of the SRAM controller address input A0 and UB, LB signals

Mode	**A0**	**iMode**	**iEN**	**oDQ0 - oDQ7**	**iDQ0 - iDQ7**
SRAM disabled	X	X	L	High-Z	High-Z
Read	L	H	H	Dout0 - Dout7	High-Z
Read	H	H	H	Dout8 - Dout15	High-Z
Write	L	L	H	High-Z	Din0 - Din7
Write	H	L	H	High-Z	Din8 - Din15

The SRAM controller has a clock input accepting not greater than a 100MHz clock signal, which is for synchronization of the control signal and data signal; in this design, all signals are triggered by the rising edge of the clock signal except iEN. When the input signal iEN is asserted LOW, the SRAM controller sets the SRAM Chip Enable (CE) pin to HIGH immediately, and the SRAM enters standby mode with minimum power consumption; meanwhile, the data terminals become high impedance. When iEN is asserted HIGH, the controller sets the $\overline{\text{CE}}$ pin to LOW, and the SRAM chip is enabled to use. The input signal, iMode, is used to determine the write or read action: when iMode is asserted HIGH, the controller reads data from the SRAM; when iMode is asserted LOW, the controller writes data to the SRAM.

In the physical connections, the SRAM controller's sADDR17 to sADDR0 pins are connected to pin A17 to A0 of the SRAM sequentially. The SRAM controller's sDQ15 to sDQ0 pins are connected to pin D15 to D0 of the SRAM. Other control signals, such as CE, OE, WE, LB, UB, are also connected to the corresponding pins on the SRAM.

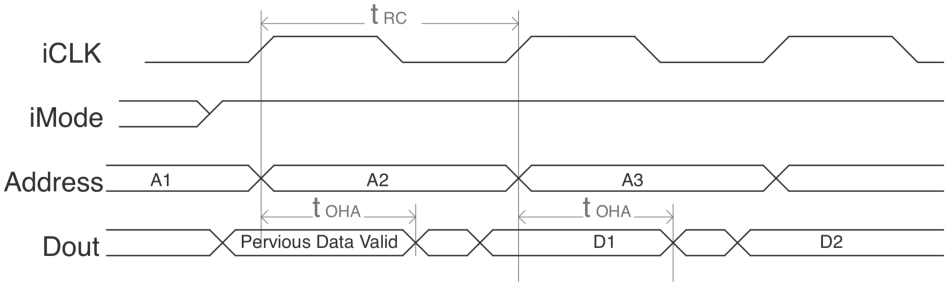


Figure 3.10: SRAM Read switch

Read access is operated when iMode is set to be HIGH. The SRAM controller maps data from the SRAM's Dout0 ~ Dout7 to oDQ0 ~ oDQ7 when the least significant bit of the controller's Address Input (A0) is 0. Conversely, when A0 is 1, the SRAM controller maps data from SRAM's Dout8 ~ Dout15 to oDQ0 ~ oDQ7. At this time, the input interfaces iDQ0 ~ iDQ7 are high impedance. The minimum value of the Read Cycle Time (t_{RC}) is 10ns. The minimum value of Output Hold Time (t_{OHA}) is 2ns; These values are part of the SRAM switching characteristics, which may be different depending on the SRAM chip used.

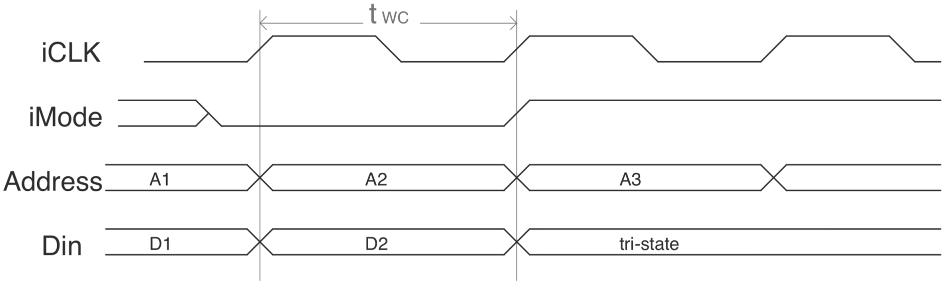


Figure 3.11: SRAM Write switch

Write access is operated when iMode is set to be LOW. The SRAM controller maps data from iDQ0 ~ iDQ7 to SRAM's Din0 ~ Din7 when the least significant bit of the controller's Address Input (A0) is 0. Conversely, when A0 is 1, the SRAM controller maps data from iDQ0 ~ iDQ7 to SRAM's Din8 ~ Din15. At this time, the output interfaces oDQ0 ~ oDQ7 are high impedance. Table xx shows the control signal and data signal in the different operating modes. The minimum value of the Write Cycle Time (t_{RC}) is 10ns which is part of the SRAM switching characteristics, and may be different depending on the SRAM chip used.

3.4.2 UART

Serial Communication is widely used in embedded system development, and one of the most commonly used protocols is Universal Asynchronous Receiver-Transmitter (UART). It has no synchronized clock signal during data transfer which makes it an asynchronous protocol. Asynchronous transmission allows data to be transmitted with timing parameters agreed by both the sender and receiver. There are three wires required to transfer data: one wire for data transmitting, one wire for data receiving, and the last one for ground reference. When transmitting data, the transmitter converts incoming parallel data to serial data and sends byte data on the communication line bit by bit. When receiving data, the receiver obtains data from the communication line bit by bit, and temporarily stores them in the memory until a whole byte data is collected, then it sends the byte length data to the processor in parallel. Using an FPGA to implement UART avoids using extra chips, thus saving space and enhancing system reliability.

Because UART is an asynchronous transmission protocol, keeping the same transmission speed between transmitter and receiver is a prerequisite for ensuring correct data transmission and reception. The baud rate is used to specify the transmission speed of both parties. Baud rate is the unit for symbol rate represented by bit per second in the UART. Data is transmitted in bytes; each byte contains 8 bits. It takes bytes of data and sequentially sends the individual bits. There is a start bit in front of the 8-bit data, and a stop bit at the end of the 8-bit data. Figure 3.12 illustrates the single byte data formatting.

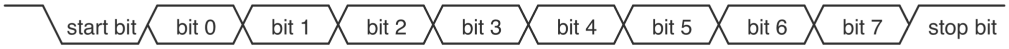


Figure 3.12: UART Frame

The FPGA UART design is modularized; according to the functions, the whole system is separated into baud rate generator, transmitter, and receiver. Figure 3.13 shows its block diagram.

When the communication line is idle, the line stays in a high-voltage state, which is considered as logic HIGH. Once the transmitter starts to send data, it will generate a start bit signal, which is represented by a low-voltage (logic LOW), on the communication line before each data byte. The baud rate determines the time required to

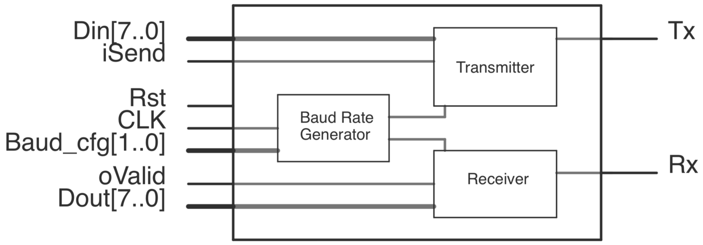


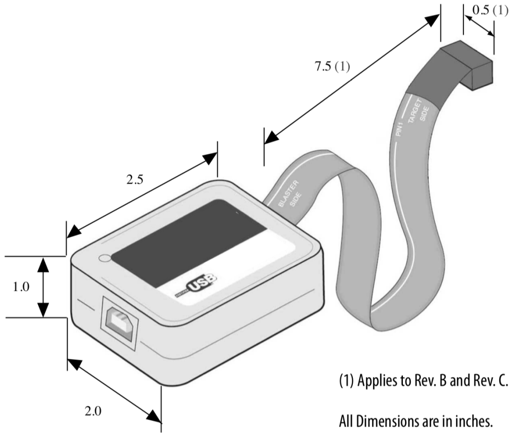
Figure 3.13: UART block diagram

transfer each bit. If the baud rate is 1, the start bit logic LOW persists for 1 second and then the logic level will change according to the value of bit 0. If the value of bit 0 is 1, the communication line becomes high-voltage. If the value of bit 0 is 0, the communication line becomes low-voltage and remains for 1 second. The transmission follows the same rules, until the last bit of the data byte is transferred, then a stop bit is generated, which is a high-voltage (logic HIGH) on the communication line. The stop bit means the receiver is informed that this is the end of the current byte transfer. After the stop bit, the communication line returns to the idle state. The stop bit can be 1-bit, 2-bit or 3-bit based on the requirement of the application environment. When the stop bit is 1-bit, a one-byte data frame consists of 10 bits; it needs 10 seconds to complete the transfer if the baud rate is 1. It requires 86.8 μs to complete the transfer if the baud rate is 115200.

When receiving data, the receiver samples the incoming data on the communication line. If the initial state is assumed to be idle on the communication line, once the receiver detects a logic LOW signal, it changes to receiving mode immediately. It samples the incoming data based on the pre-set baud rate and stores each bit sequentially. After collecting all the bits of the data ending up with a stop bit, the data transfer is completed. The receiver continues monitoring for a change of logic level on the communication line.

3.5 Configuring FM10 with USB-Blaster

Intel MAX10 series FPGAs can be configured using a JTAG connector. Intel provides users with a USB-Blaster that can send configuration data from the host computer to the FPGA. It has a USB connector and a standard 10-pin female connector, in which the USB connector connects to a host computer and the other end connects to the FPGA's JTAG interface located on the PCB. Developers can use Intel Quartus Prime software to compile the VHDL code, and then the compiled file can be uploaded to FPGA through a USB-Blaster to finish the FPGA configuration process.

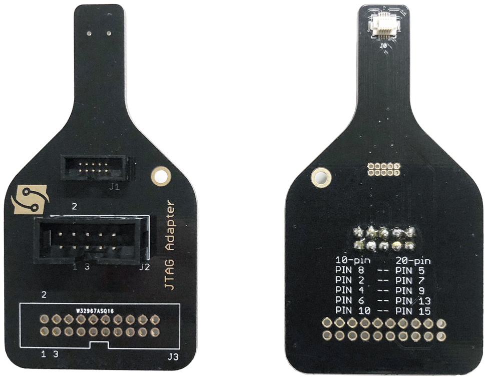


(1) Applies to Rev. B and Rev. C.

All Dimensions are in inches.

Figure 3.14: USB Blaster

However, the size of the 10-pin female connector on the USB Blaster is relatively big; it is 12.7 mm in length, which is against the compact design intention of FM10, meaning there is not enough space for the mating connector. FM10 substitutes it with a DF12(3.0)-10DP-0.5V(86) female connector from Hirose Electric Co Ltd, which has the same number of pins but is smaller in size (5.5mm x 3.8mm). In order to establish the connection between the 10-pin connection header on the USB-Blaster and the Hirose connector on FM10, a JTAG adapter was designed. Figure 3.15 shows the JTAG Adapter.



a) Top View

b) Bottom View

Figure 3.15: JTAG Adapter

The JTAG Adapter has a DF12 (3.0)-10DS-0.5V(86) male connector that is the mating connector to the female connector on FM10. Meanwhile, it also has a standard 10-pin male connector that the USB-Blaster can connect to. The Hirose male connector and the standard 10-pin header connector are interconnected correspondingly via traces in the PCB. The JTAG Adapter acts as a bridge to connect the USB-Blaster and the JTAG of FM10. Figure 3.16 shows the signals on the USB-Blaster 10-pin connector and the design of the FM10 JTAG.

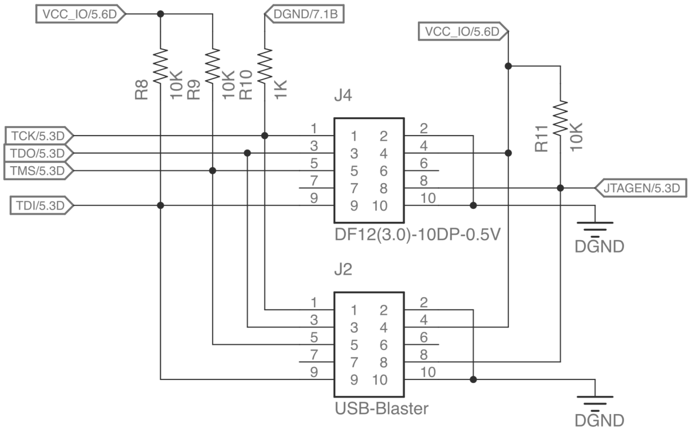


Figure 3.16: The JTAG adapter schematic

3.6 Conclusions

The chapter proposes a compact-sized FPGA processing module with stackable interface offers Hard-Real-Time parallel digital logic computing capabilities. The study demonstrates 6 layers stack PCB design, power management, connections between SRAM and FPGA chip, FTDI, CAN transceiver, and the design of stackable interface.

The SRAM controller was implemented in VHDL, it can facilitate user access to data stored in SRAM, control the mode of SRAM between read and write status. The design of SRAM controllers demonstrates converting a 16-bit data with 18-bit wide address SRAM to a 19-bit wide address SRAM for 8-bit wide data.

The FM10 module can work standalone, it can also collaborate with multiple FM10 or EM4 modules via the stackable connector. In chapter 4, the EM4 module is introduced.

Chapter 4

Design and Implementation of the MCU Module

4.1 EM4 Board Overview

The EM4 is a compact microprocessor processing module among the ioNeurons^{TM} series, which allows users to develop applications with STMicroelectronics STM32F427ZIT7 high-performance microcontroller efficiently. Additionally, the EM4 is designed to be compatible with Intel® Edison and ioNeurons^{TM} FM10 in hardware interfaces, so it has flexible compatibility and scalability, which means it can be applied in complex computing environments. The Intel® Edison is a single-board computer powered by Intel® Atom SoC dual-core CPU; it allows users to run programs based on Linux or Windows operating system; Compared to Intel® Edison, running programs without any operating system on the EM4 has much better real-time performances. On the other hand, it can free up the Intel® Edison for higher level tasks. The EM4's microprocessor features a Floating Point Unit (FPU) single precision that helps FPGA accomplish complicated floating point calculation avoiding occupying excessive logic gates resources. Interpreting the overall platform structure of ioNeurons$^{\text{TM}}$, not only does the EM4 compensate the work that Intel$^{®}$ Edison and FPGA are not good at, but also it extends the computing capacity of the platform, which entails the win-win effect of the design.

The EM4 module incorporates high-speed embedded memories which are 256 Kbytes Static Random Access Memory (SRAM) and 64 Mbytes Synchronous Dynamic Random Access Memory (SDRAM); there is a 2 Mbytes Flash memory available for storing programs and data. A Micro SD card socket is shared by STM32F427 microcontroller and Intel$^{®}$ Edison to expand storage spaces.

The Integrated 3.3V to 16.0V DC power supply and power over USB are regulated via a highly efficient power management module, which also provides 3.3V and 5.0V DC power source with maximum 1.0A current to power peripherals. The EM4 is also capable of powering the Intel$^{®}$ Edison.

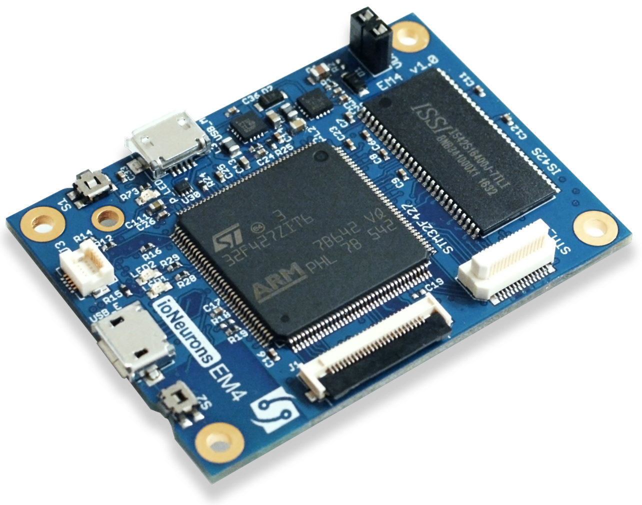


Figure 4.1: Top view of ioNeurons$^{\text{TM}}$ EM4

The EM4 supports standard communication interfaces, such as Inter-Integrate Circuit (I2C), Universal Asynchronous Receiver Transmitter (UART), Serial Peripheral

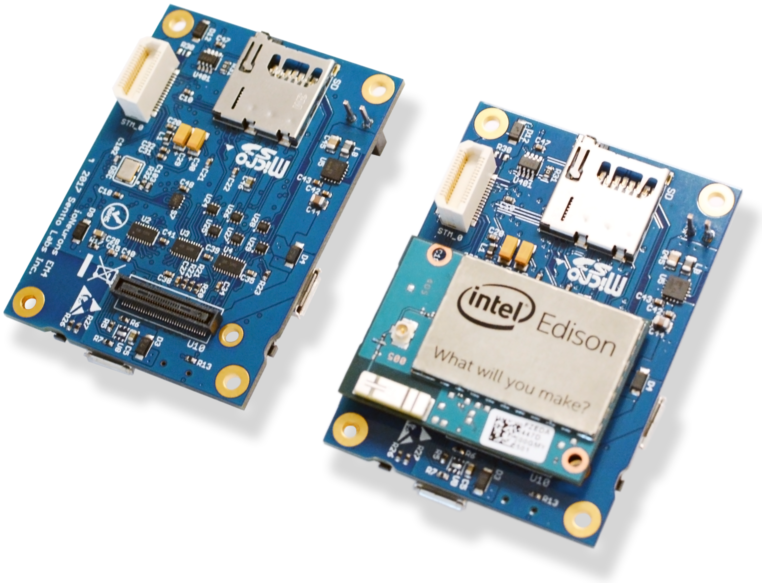


Figure 4.2: Bottom view of ioNeurons$^{\text{TM}}$ EM4 with Intel$^{\circledR}$ Edison

Interface (SPI), and it integrates Controller Area Network (CAN) transistor and USB type-AB connector for USB On-The-Go with full-speed capacity. For analog signal sampling, it offers three 12-bit Analog-to-Digital Converter (ADC), and two general purpose 32-bit timers.

The physical dimension of EM4 is 40.0 mm x 55.0 mm precisely same as ioNeurons$^{\text{TM}}$ FM10 module. It has a 24-pin Flat Flex Cable (FFC) connector and two 30-pin P5K series connectors, which permit 40 GPIOs in total and outputs of 3.3V or 5.0V voltage individually. The EM4 is equipped with user programmable LEDs, and push buttons. The EM4 can work standalone; however, with the design of the stackable connector, it can also collaborate with multiple FM10 or EM4 modules, which can enhance the computing capacity and increase the number of Input/Output interfaces.

4.2 Layout and Components

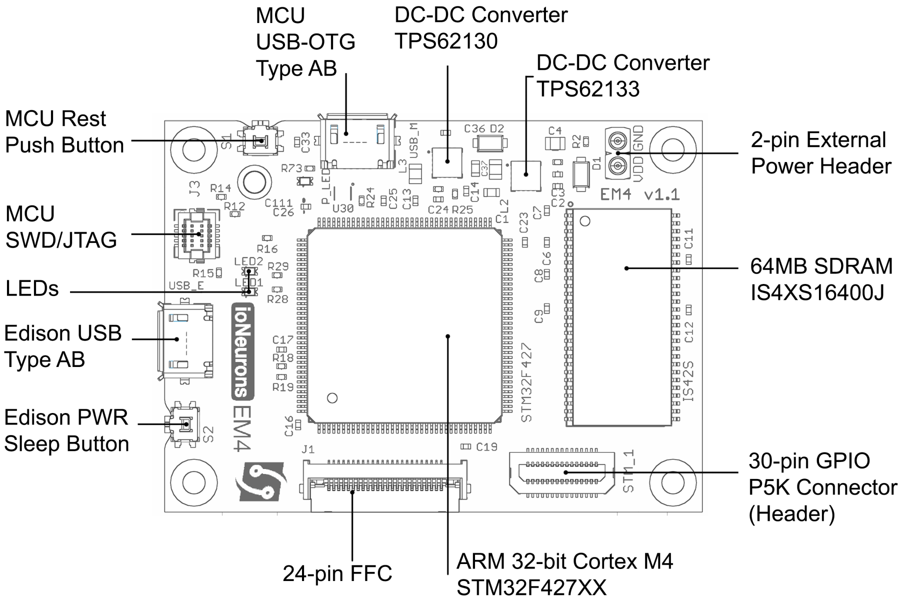


Figure 4.3: EM4 Board PCB and component diagram (top view)

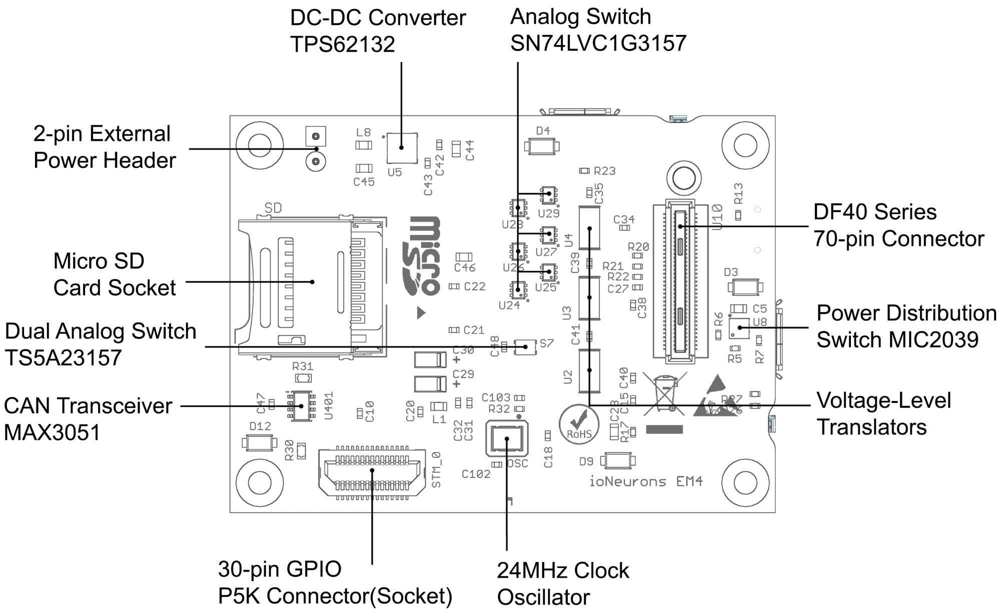


Figure 4.4: EM4 Board PCB and component diagram (bottom view)

The type and quantity of electronic components required to make up the EM4 module are much more than what FM10 needs. It is considerably challenging to integrate these electronic components into a same-sized PCB. It is vitally important that every component is placed in a suitable position on a printed circuit board since it determines the overall dimension, number of layer, and complexity of tracing. There are limited options for allocated space of FFC connector and P5KS series connectors. When the FM10 module and the EM4 module are stacked together, the FFC connectors should not block each other, and the P5K series connectors of the EM4 must correspond to the FM10's P5K series connectors to achieve the physical stack-able structure.

Taking production costs and manufacturing techniques into account, the EM4 employs a 6 layer PCB structure and plating through holes only; no blind via holes or buried via holes were used since they significantly increase the cost of production. The minimum diameter of through holes is 0.15 mm which is usually the smallest diameter achievable with mechanical drilling. The solder resist layer covers any through holes which are under 0.35 mm in diameter to protect them and avoid a short circuit, and the stop layer mask is not applied to these hole locations. The minimum width of a trace is 0.1524 mm, and the minimum clearance between different signal wires is 0.1778 mm. The minimum distance between objects on signal layers and the board dimension is 0.2032 mm, and the thermal isolation is 0.1778 mm. These parameters are constrained by the production techniques of PCB manufactures.

The FR4 insulation material is applied between any two conductor layers. Table 4.1 shows the PCB layer stack. Electroless Nickel Immersion Gold (ENIG) surface treatment is implemented on the top and bottom layer during PCB manufacture to prevent the nickel from oxidizing with a thin layer of immersion gold.

When determining the relative position of components, the one having the most pins takes priority because its position has the most significant impact on tracing. The

Table 4.1: EM4 PCB Layer Stack

Layer	**Layer Stack**	**Layer Name**	**Material**	**Thickness**	**Plane**
1		Top paster	Sn63/Pb37	0.12mm	
2		Top overlay			
3		Top solder	Solder resist	0.01mm	
4	- - - - - - - - - - - - -	Top ENIG	Gold	0.05μm	
5	— — — — — —	Top	Copper	0.018mm	Signal
6	///////////////	Dielectric 1	FR-4 TG140	0.0913mm	
7	— — — — — —	Layer 2	Copper	0.03mm	D_GND
8	□ □ □ □	Dielectric 2	FR-4 TG140	0.53mm	
9	— — — — — —	Layer 3	Copper	0.03mm	Signal
10	///////////////	Dielectric 3	FR-4 TG140	0.1056mm	
11	— — — — — —	Layer 4	Copper	0.03mm	5V_SYS
12	□ □ □ □	Dielectric 4	FR-4 TG140	0.53mm	
13	— — — — — —	Layer 5	Copper	0.03mm	A_GND
14	///////////////	Dielectric 6	FR-4 TG140	0.0913mm	
15	— — — — — —	Bottom	Copper	0.018mm	Signal
16	- - - - - - - - - - - - -	Bottom ENIG	Gold	0.05μm	
17		Bottom solder	Solder resist	0.01mm	
18		Bottom overlay			
19		Bottom paster	Sn63/Pb37	0.12mm	

MCU chip (STM32F427ZIT7) has 144 pins; it is the chip with the largest number of pins on this board. Relatively SDRAM has lots of pins as well and needs to establish connections with MCU. The advanced communication interface connecting MCU and SDRAM is Flexible Memory Control (FMC) interface; most of the FMC functional pins are located between pin 37 to pin 108 of the MCU chip. If the edge containing pin 1 to pin 36 is defined as the bottom side of the MCU chip, the SDRAM should be placed on the right of the MCU to reduce the length and difficulty of PCB traces. In addition, the high-speed external clock a 27 MHz crystal is placed as close as possible to the oscillator pins of MCU in order to minimize distortion and startup stabilization time. This high-frequency clock signal is a critical factor to maintain stable operation of the whole system.

In order to reduce electromagnetic interference (EMI), the EM4 6-layer structure adopts the top layer, the third layer, and the bottom layer as signal layers, in which copper is paved in the area around signal routes and connects to the ground plane. The second and the fifth layer are ground planes, and the fourth layer is the power plane. The solid ground plane will minimize inductance, and any signal plane close to the ground plane will have continuous return current. Decoupling capacitors are placed near the power pin of chips. When the chip's payload suddenly increases, the circuit tries to increase its current; however, the inductance in the power supply line hinders the current rising and causes the power line voltage to sag. At this time the decoupling capacitors provide short bursts of current to maintain the voltage at a stable level. In the actual design, two external capacitor 2.2uF were connected to the Vcap1 and Vcap2 pins of the MCU chip to achieving the stabilization of the main regulator.

4.2.1 Power management and distribution

The EM4 module can be powered by a DC power supply, USB, and Li-Ion or other types of battery because of its high-performance power management module design, which allows EM4 to work in various application environments with flexible power requirements easily. At the same time, the 5.0 V and 3.3 V are power outputs from the EM4 module that sources a maximum of 2.0 A total, which allows small sensor modules that connect to EM4 to be powered, thus avoiding redundant power circuit design for external sensors.

There are 7 power rails on the EM4: Vin, USB_VBUS, VSYS, 3.3V, 4.38V, 5V, 1.8V. The Vin and USB_VBUS are the only input power rails to the EM4 module. The voltage range of Vin is 5.0 V to 17.0 V; the USB_VBUS is the standard USB voltage, and the range is from 4.75 V to 5.25 V. The VSYS is the system voltage that is used to power microcontroller at fixed 3.3 V. The 4.38 V is the input voltage to power Intel® Edison. The 1.8 V is power output from Intel® Edison module that sources a maximum of 100 mA, which is used to power voltage-level translator chips. Figure 4.5 shows the power tree of EM4.

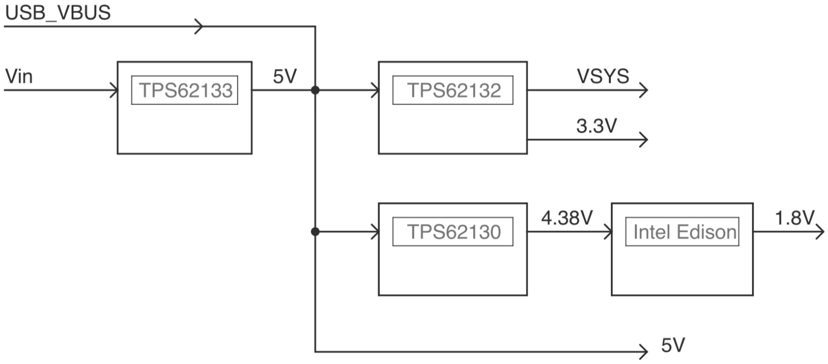


Figure 4.5: The Power Tree of EM4 Module

The EM4 power management module contains three synchronous Step-Down DC to DC Converters from Texas Instruments; their part numbers are TPS62133, TPS62132, and TPS62130. All of them adopts the 16-pin 3x3 mm QFN package and can provide 3A of continuous output with a maximum converting efficiency up to 95%. These chips are made for mobile application platforms.

The TPS6213X DC-DC converters use advanced regulation topology (DCS-Control) that combines advantages of multiple control modes, which can monitor output voltage changes and feed it directly to a fast comparator to set the switching frequency. This technology provides immediate response for dynamic load changes to achieve fast and stable voltage operation. These DC to DC converters also prevent the circuit from shorting and overloading. When a load or a short circuit is detected and the output voltage drops below 0.5 V, the current limit will be reduced to 1.6 A immediately. If the output voltage picks up above 0.5 V again, the device recovers to the normal operating mode. A 1.0 uH inductor and a 22.0 uF capacitor are connected to converter's output as an external LC output filter. Figure 4.6 , Figure 4.7, and Figure 4.8 shows the schematics of each DC-DC voltage converter connectivity implemented in the EM4.

All TPS6213X chips accept a maximum of 17.0 V input power. The TPS62133 has a fixed output voltage 5.0 V which is the input power of the TPS62132 and the TPS62130 chip. The TPS62132 provides a fixed 3.3 V output voltage to the MCU and other chips whose rated voltages are 3.3 V. The output voltage of the TPS62130 is adjustable from 0.9 V to 6.0 V by using a resistive divider between output voltage and ground. In the EM4 design, the TPS62130 outputs 4.38 V to power Intel Edison whose required input voltage range is between 3.15V to 4.5V. A DB2W900L diode (D2) from Panasonic Electronic Components is directly placed after the output of TPS62133 chip to prevent reverse current. Moreover, the EM4 has two USB interfaces, in which the power rail joins to the 5.0 V power rail through DB2W900L diodes (e.g. D3). Design mentioned above allows the system to be powered by the USB connectors directly. All diodes feature 30.0 V reverse breakdown voltage, which prevents the USB from being burnt by a high reverse voltage, and allows FM10 to be powered by batteries and USB at the same time. Besides, these diodes are further used in 3.3 V and 5.0 V output power rail assuring the unidirectional conductivity. Users cannot use these output power rails to power the system; this design prevents the chips from burning because of supplying the unstable or inaccurate supply voltage.

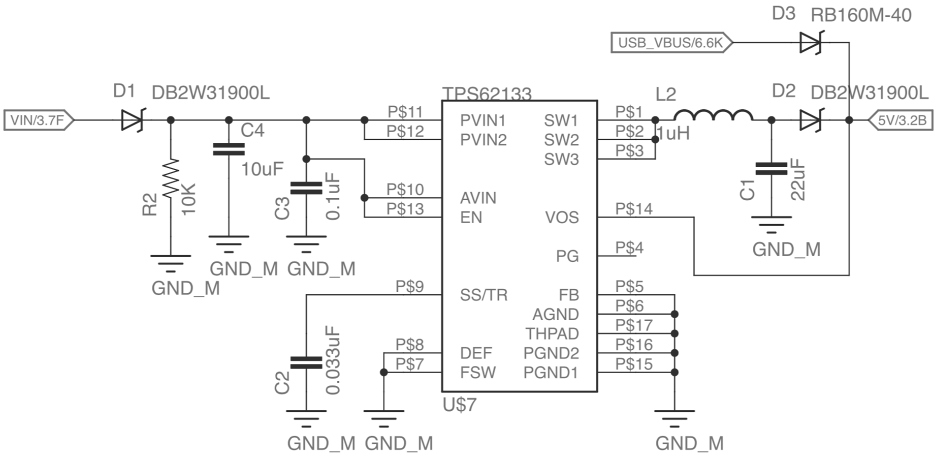


Figure 4.6: EM4 module power schematic: TPS62133

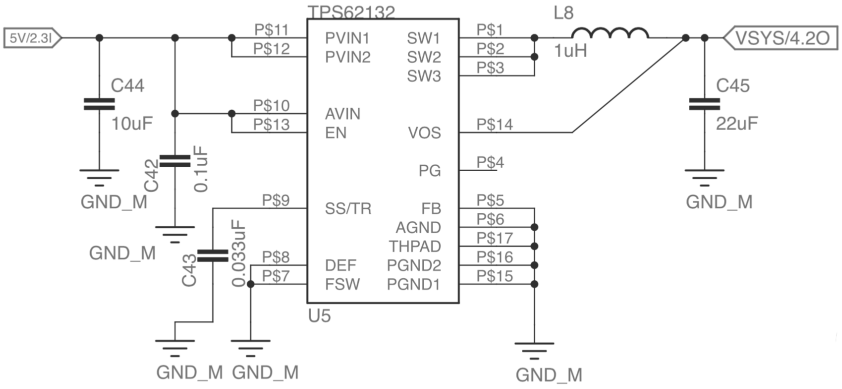


Figure 4.7: EM4 module power schematic: TPS62132

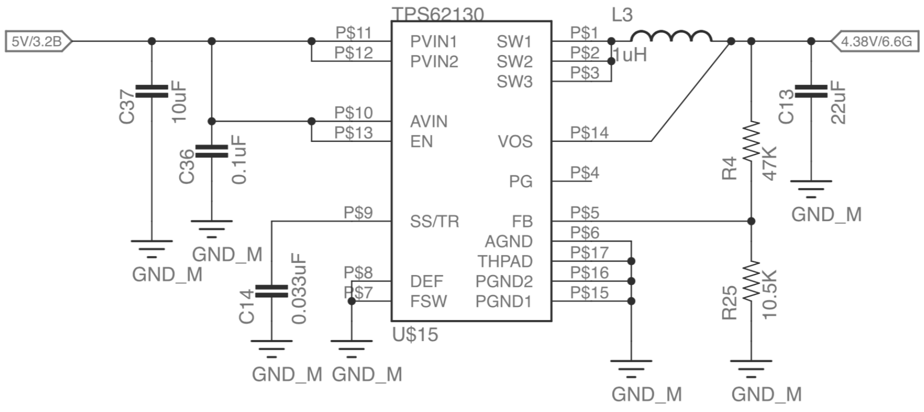


Figure 4.8: EM4 module power schematic: TPS62130

4.2.2 MCU: STM32F427ZIT7

The chip, STM32F427ZIT7, belongs to STMicroelectronics STM32F4 family. It is the high-performance ARM Cortex -M4 32-bit core microcontroller operating at a frequency up to 180Hz. It is a single-supply MCU, which requires only one external power supply of 3.3V.

The STM32F427XX embeds a flexible memory controller (FMC) interface, which is connected to a 64MB SDRAM with 16-bit data bus width in the EM4 design. It also integrates 2Mbytes flash memory for storing programs and data. A 256 Kbytes system SRAM is embedded, which includes 4 Kbytes backup SRAM. The 4Kbytes SRAM can be used to store data even in the standby mode.

The EM4 uses an EQFP package STM32F427XX chip, offering 144 pins, in which 115 pins have general-purpose input/output (GPIO) functionality. Users can program it as an input or output. Most of the GPIO pins serve digital or analog alternate functions. For digital functions, it supports standard and advanced communication interfaces such as I2C, SPI, UART, CAN, and USB OTG. In the stackable structure, the way of how to use the I2C and CAN interfaces is particularly important. Becuase they are able to make the microcontroller as a master or slave node in the communication network, to achieve communication between multiple devices. The I2C supports standard mode with data rates up to 100Kbps, and fast-mode with data rates up to 400Kbps. The CAN interface supports 1Mbit/s bit rate. For analog functions, there are three 12-bit analog-to-digital (ADC) converters, and each ADC can be shared by 16 external channels. In EM4, some of the ADC channels are no longer available since the pins have been assigned to other electronic components. The STM32F427XX is fully pin-to-pin, software and feature compatible with STM32F429XX. Comparing with STM32F427XX, STM32F429XX has a unique LCD-TFT controller that provides a 24-bit parallel digital RGB digital signal to a broad range of LCD and TFT panels.

STM32F427XX microcontroller supports three low-power modes: sleep mode, stop mode and standby mode. The stop mode achieves the lowest power consumption when CPU, clock, and PLLs stop working and retain the data in SRAM and registers. It can be waked up by RTC alarm, wakeup events or USB OTG FS/HS. The standby mode also achieves the lowest power consumption, but the data in SRAM and registers would be lost except the registers in the backup domain and the 4KB backup SRAM when enabled. The device can be waked up through NRST pin, WKUP pin, and RTC alarm. In sleep mode, only the CPU does not work; interrupts or events can wake it up. The three different low-power modes can be adopted in various application environment as need.

STM32F427XX embedded a 16 MHz internal RC oscillator as the default CPU clock. However, the accuracy of internal RC oscillator would be affected when the CPU temperature varies, thus diminish the accuracy of high-speed serial protocol communications. In order to overcome this defect, a 24MHz crystal is integrated on the EM4 module to generate more accurate and stable clock signal regardless temperature changes. This external clock signal can be monitored for failure. When failure is detected and the external clock signal does not work properly, the system will switch to the internal RC oscillator automatically. EM4 adopts NX3225SA quartz crystal unit from NDK American to implement the external oscillator design, which requires two 8pF load capacitors. To minimize the output distortion and start-up stabilization time, the resonator and the load capacitors should be placed as close as possible. Figure 4.9 shows the high-speed external clock schematic.

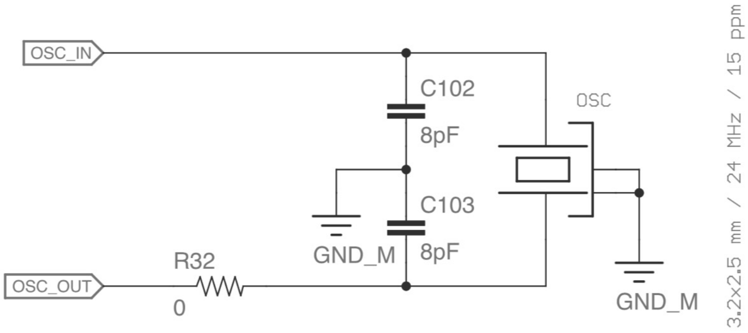


Figure 4.9: EM4 OSC schematic: TPS62130

STM32F427XX has a pre-programmed boot loader that is stored in the internal system memory. Its primary job is to load the programs to program memory in the processor. The bootloader firmware of STM32F4 series support user flash, and internal SRAM. Developers can access the bootloader mode by setting up the signal on the boot0 pin. In EM4, the boot0 pin connected to a pull-down resistor. When boot0 = 1, the device enters bootloader mode, developers can program the device by using UART, SIP or USB DFU methods.

STM32F427XX integrates Serial Wire JTAG Debug Port (SWJ-DP) that is a combined JTAG and serial wire debug port. In EM4, this port is connected to the 10-pin SWD/JTAG user interface which provides a connection interface for ST-Link debugger. ST-link can help users to program the microcontroller and provide the USB full-speed interface to communicate with ARM® mbed, Atollic®, or Keil® development environment.

4.2.3 SDRAM: IS42S16400J

Synchronous Dynamic Random Access Memory (SDRAM) is a kind of random access memory. It has a synchronous CLK input interface. To synchronize with the system bus of CPU, it would wait for a rising edge of the clock input before taking any actions to the incoming control signals. The internal command execution and data input/output are all based on the synchronous CLK signal.

The SDRAM is also volatile memory and would lose the stored data after powering off; this property is the same as SRAM. The distinction about SDRAM is that it uses a single capacitor and a single transistor to store one bit. The capacitor will be charged when the transistor is turned on. However, the capacitor leaks off energy. The data in SDRAM must be refreshed periodically via regularly read or re-written to ensure the effectiveness of the data, which implies the meaning of dynamic. The refresh action is the most critical operation and directly relate to the performance of SDRAM. The structure of SDRAM internal data storage circuit is much simpler than that of SRAM, and it offers more storage space in the same chip size. If ioNeurons^{TM} FM10 needs to store and process a significant amount of data, the 4 Megabit SRAM can be replaced by IS42S16400J SDRAM in case the SRAM can not meet the demand for more storage.

IS42S16400J provides 1048576 bits x 16-bit x 4-bank, for a total of 64 Megabit storage space. All input and output signals refer to the rising edge of the input clock. This SDRAM was chosen for the EM4 design because it offers auto refresh feature as results of integrating a complex refresh controller, which frees the CPU from redundant work and eliminate the developers' concern about refreshing cycle calculation. The refresh controller performs refresh operation 4096 times every 64 ms. During a self-refresh operation, all data and address inputs pins of IS42S16400J are in 'Do not care' state and the CKE pin remains LOW. The next command cannot be executed until the self-refresh operation is completed. When a self-refresh operation finished, the CKE pin goes HIGH, then read and write operation are resumed. Figure 4.10 shows pin map of IS42S16400J.

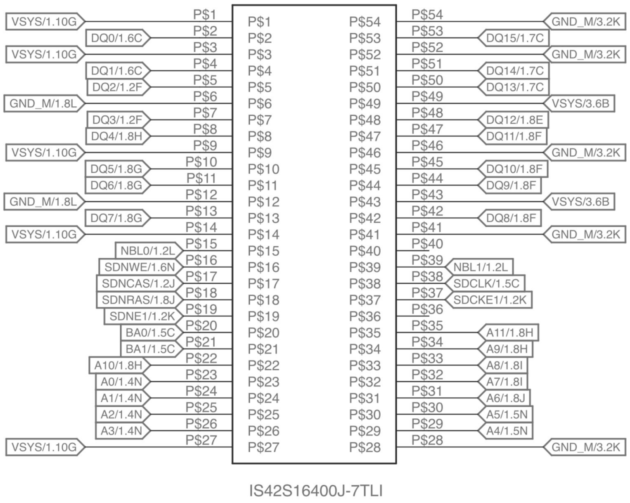


Figure 4.10: EM4 SDRAM schematic: TPS62130

4.2.4 On board external Micro SD card slot

The EM4 embedded a Micro Secure Digital (SD) card slot that allows data to be accessed through a Micro SD card as an external data storage for the ioNeurons^{TM} system, with storage space scalable to gigabytes. The Micro SD card is widely used in data storage of portable devices. The ability of EM4 to support SD card is vitally important for applications which requires long-term data acquisition and recording applications.

In EM4, the micro SD card slot is designed to be shared by both STM32F427ZIT7 and Intel® Edison. This design increases the way of data exchange between two modules, and the storage space of each module is expanded. The EM4 implements dual data channels using single-pole double-throw (SPDT) analog switches to share the SD card slot. Figure 4.11 shows the block diagram of the shard SD card slot design.

The SPDT switch on EM4 is SN74LVC1G3157 that comes from Texas Instruments, which can handle analog and digital signals. It has two switch IOs (B1, B2) and a common terminal (A). The SD card slot's interfaces are connected to common terminals. The microprocessor STM32F427ZIT7 determines whether the signals of the common terminals come from STM32F427ZIT7 or Intel® Edison by controlling the "select" signals of the SPDT switches. The entire switch circuit contains 6 SPDT switches; their "select" control signals are connected in parallel and finally connected to pin PG6 of STM32F427ZIT7. When the control input 'select' is HIGH, the B2 channel is selected, and it is connected with common terminal A. When 'select' is set to LOW, B1 and A are connected. The signals can be transmitted in either direction.

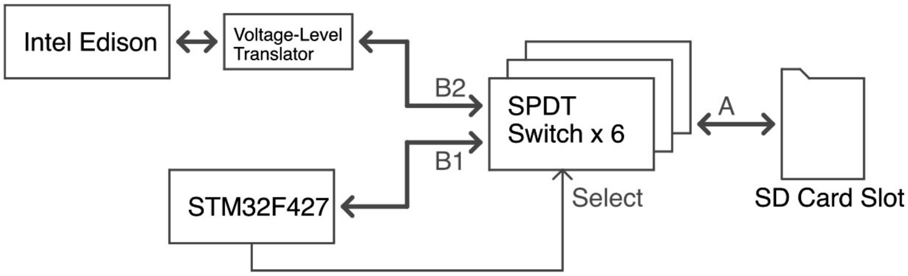


Figure 4.11: Block diagram of shared SD card slot

Based on SD specification, SD cards operate at the standard 2.7 3.6V, while the input and output interface voltages of Intel Edison are 1.8 V transistor-transistor logic (TTL) levels. In order for Intel® Edison to access the SD card and performing read/write operations, the Intel® Edison's interface voltage level of must be converted to 3.3 V TTL level before reaching the SD card, and the SD card output signals voltage level must be dropped to 1.8 TTL level before entering Intel Edison. In the EM4 design, Texas Instruments TXB0108 8-bit Bidirectional Voltage-Level translator takes responsibility of converting the signal voltage on the Intel® Edison interfaces. Figure 4.12 illustrates the schematic of the shared Micro SD card slot. As shown

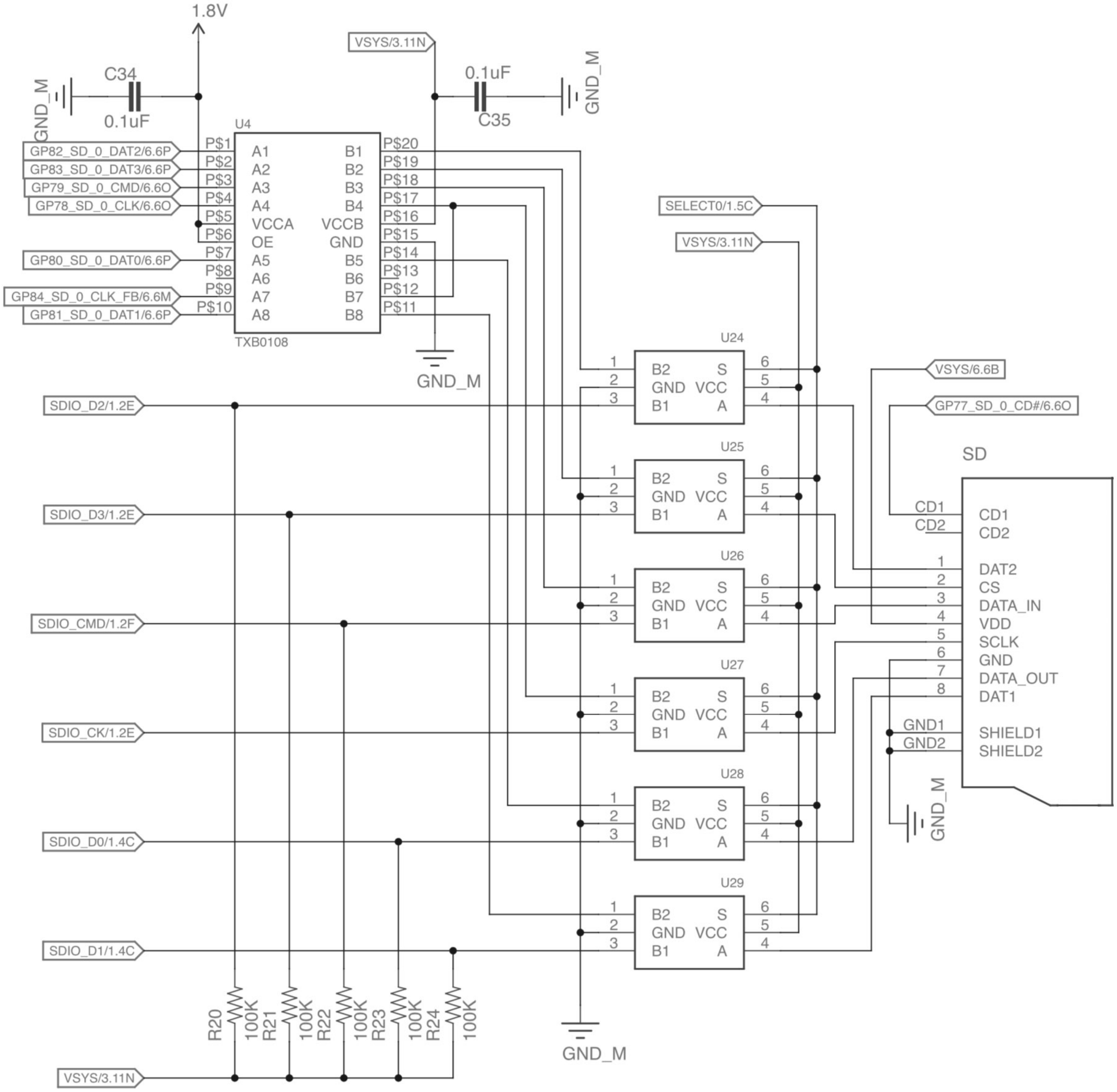


Figure 4.12: The schematic of EM4 shared SD card slot

in the figure, a total of 7 port signals from Intel® Edison need to be converted. A TXB0108 chip can complete the voltage translation for all signals because it includes 8 channels. TXB0108 contains two separate configurable power-supply rails. The A port is powered by 1.8V and connected to Intel® Edison secure digital input/output interface. The B port is powered by 3.3V, and connected to the B2 port of the SPDT switches. The output-enable (OE) input is locked at HIGH, which means the translator is always enabled. TXB0108 does not require a direction control signal because it is a directionless voltage-level translator.

4.2.5 CAN Transceiver

The EM4 embedded CAN transceiver which is MAX3051 from Maxim Integrate. A CAN bus standing for Controller Area Network bus is a robust data communication protocol that was initially used on vehicles enabling devices to communicate with each other, but now it is not limited to use in the automobile field only. CAN's high performance and reliability have been recognized, and are widely used in industrial automation, ships, medical equipment, and so on. The CAN transceiver makes the EM4 interfaces more friendly to access CAN network, and this greatly increases the applications of EM4. The CAN bus provides powerful support for distributed control platform which is using ioNeurons^{TM} processing modules, to realize real-time and reliable data communication between various nodes.

Unlike other data transfer protocols, a pair of CAN bus twisted wires allows more than two devices to connect in parallel and exchange data; in other words, it is not a fixed point-to-point communication protocol; it can create a communication network and offer flexible point-to-point or point-to-multipoint communications. Those devices connecting to a CAN bus are Electronic Control Units (ECUs) also known as nodes. A node can be a simple I/O device or a complicated embedded computer. Any of the devices can be a host as long as it has the corresponding functionality. Once a device broadcasts a message on the bus, every other device will listen and identify if they are interested in this message, and then the message will be received by the one that is interested.

To transfer data, CAN bus signalling uses differential signals on the two wires: one is called CAN_high, the other one is called CAN_low. When CAN_high is driven to 5 V and CAN_low is driven to 0V, the transmission is recognized as dominant state representing logic level 0. When neither of them is driven, they are levelled to the reference voltage which is 2.5 V, and at this time, the transmission is recognized as recessive state representing logic level 1. In the physical layer, these two wires are twisted together in order to reduce the influence of electromagnetic interference. Two 120 Ω termination resistors are placed on each end of the CAN bus to suppress signal reflections.

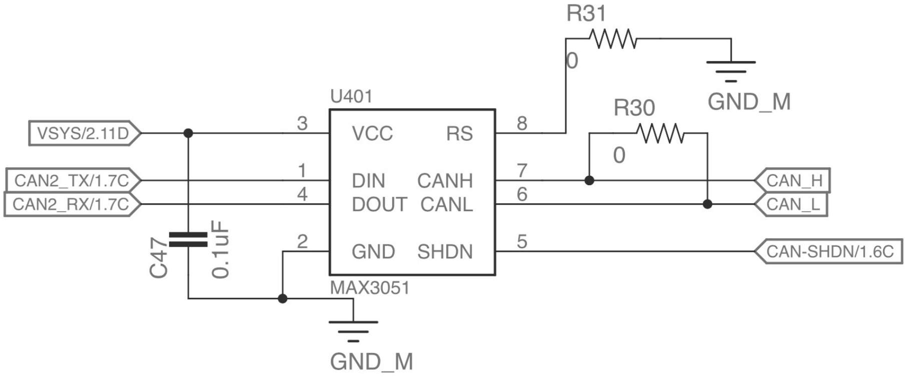


Figure 4.13: The schematic of CAN transistor in EM4 module

To generate the differential signals, a transceiver chip that is based on CAN bus protocol is required, which provides interfaces between the CAN protocol controller and the physical wires of the bus lines. In EM4, the CAN protocol controller is the microcontroller STM32F427ZIT7, which can receive and transmit a standard frame with 11-bit identifiers. The transceiver chip is MAX3051 from Maxim Integrated.

This chip can translate the logic level signal LOW or HIGH from the host controller to differential signals that are transmitted on the CAN bus. It also features four different modes of operation that enables data transfer speed control. The schematic in Figure 4.13 shows the CAN operating circuit of EM4. The Din pin of the MAX3051 connects to pin PB13 of the STM32F427ZIT7, one of the alternate function of pin PB13 is CAN2_TX. The Dout of MAX3051 connects to STM32F427ZIT7's pin PB12 whose alternate function is CAN2_RX. The pin PD12 of STM32F427ZIT7 connects to SHDN which is the control pin of MAX3051. When a HIGH state is detected on pin PD12, MAX3051 enters the shutdown mode.

4.3 Interfaces and pin mapping

As a processing module, the selection of suitable connectors for EM4 directly impacts the user experience and the way cooperated with FM10 and Intel® Edison. In order to facilitate adoption of EM4 in as many projects as possible and considering the cost-benefit factor, EM4 is designed to use three different types of connector: Flat Flex Cable (FFC) connectors, P5KS series board to board narrow pitch connectors and Hirose DF40 Series connector; All of them are classified as surface mount type and high-speed connectors. The primary function of FFC is to provide and expand the connectivity to other peripherals. The P5KS connectors are utilized for implementing expandable and stackable features between ioNeurons™ modules. The Hirose DF40 Seires connector provides interfaces and power to the Intel ® Edison.

The FFC connector mounted on EM4 module is 24-pin with a 0.4mm pitch, fitting a 12mm wide flat flex cable, which can connect EM4 with another module. In the 24-pin FFC connector, 20 out of 24 pins are defined as data transfer pins, and other 4 pins are set as power pins: 2 for VCC and 2 for GND. This reinforced dual power pin design effectively avoids the possibility of burning the connector due to overloading current on a single pin, thereby increasing the capacity to take and supply large current flow. In addition, the FFC connector has a double-bottom-contact terminal design which can remove dust and contaminants better than single-bottom-contact terminals. Two contact points on each pin also provides stable redundant contact forces for secure electrical reliability. The FFC connector is placed at the edge of the PCB; it is very close to MCU Pin_1 ~ Pin_36. This group of pins provides SPI and UART interfaces that are connected to the FFC connector. Not only does this layout design shorten the trace length between MCU and FCC, but it also reduces the EMI noise. EM4 chooses to use the parallel-to-board type FFC connector to achieve more space between two adjacent boards thus making it easier for users to install cables.

Table 4.2: EM4 24-pin FFC pin-out and signal list

FFC Pin Number	**MCU Pin Number**	**Edison Pin Number**	**Pin Name**	**Function**
1	PIN_1		PE2	TRACECLK, SPI4_SCK, SAI1_MCLK_A, ETH_MII_TXD3, EVENTOUT
2	PIN_3		PE4	TRACED1, SPI4_NSS, SAI1_FS_A, EVENTOUT
3	PIN_4		PE5	TRACED2, TIM9_CH1, SPI4_MISO, SAI1_SCK_A, EVENTOUT
4	PIN_5		PE6	TRACED3, TIM9_CH2, SPI4_MOSI, SAI1_SD_A, EVENTOUT
5	PIN_7		PC13	EVENTOUT, TAMP_1
6	PIN_129		PG14	SPI6_MOSI, USART6_TX, ETH_MII_TXD1, ETH_R_MII_TXD1, EVENTOUT

Continued on next page

Table 4.2 – *Continued from previous page*

FFC Pin Number	**MCU Pin Number**	**Edison Pin Number**	**Pin Name**	**Function**
7	PIN_128		PG13	SPI6_SCK, USART6_CTX, ETH_MII_TXD0, ETH_R_MII_TXD0, EVENTOUT
8	PIN_126		PG11	ETH_MII_TX_EN, ETH_RMII_TX_EN, EVENTOUT
9	PIN_124		PG9	USART6_RX, EVENTOUT
10	PIN_76		PB15	TIM1_CH3N, TIM8_CH3N, SPI2_MOSI, I2S2_SD, TIM12_CH2, OTG_HS_DP, EVENTOUT
11				GND
12				GND
13				VCC 3.3V
14				VCC 3.3V
15	PIN_27		PC1	ETH_MDC, EVENTOUT, ADC123_IN12
16	PIN_18	PIN_53	PF6, GP110	TIM10_CH1, SPI5_NSS, SAI1_SD_B, UART7_RX,EVENT_OUT, EDISON_SPI2_FS0, EDISON_GPIO
17	PIN_19	PIN_55	PF7, GP109	TIM11_CH1, SPI5_SCK, SAI1_MCLK_B, UART7_Tx, EVENTOUT, ADC3_IN5, EDISON_SPI2_CLK, EDISON_GPIO
18	PIN_20	PIN_59	PF8, GP114	SPI5_MISO, SAI1_SCK_B, TIM13_CH1, EVENTOUT, EDISON_SPI2_RXD, EDISON_GPIO
19	PIN_21	PIN_57	PF9, GP115	SPI5_MOSI, SAI1_FS_B, TIM14_CH1, FMC_CD, EVENTOUT, EDISON_SPI2_TXD, EDISON_GPIO

Continued on next page

Table 4.2 – *Continued from previous page*

FFC Pin Number	**MCU Pin Number**	**Edison Pin Number**	**Pin Name**	**Function**
20	PIN_45		PC5	ETH_MII_RXD1, ETH_RMII_RXD1, EVENTOUT, ADC12_IN15
21	PIN_44		PC4	ETH_MII_RXD0, ETH_RMII_RXD0, EVENTOUT, ADC12_IN14
22	PIN_47		PB1	TIM1_CH3N, TIM3_CH4, TIM8_CH3N, OTG_HS_ULPI_D2, ETH_MII_RXD3, EVENTOUT, ADC12_IN9
23	PIN_75		PB14	TIM1_CH2N, TIM8_CH2N, SPI2_MISO, I2S2ext_SD, USART3_RTS, TIM12_CH1, OTG_HS_DM, EVENTOUT
24	PIN_46		PB0	TIM1_CH2N, TIM3_CH3, TIM8_CH2N, OTG_HS_ULPI_D1, ETH_MII_RXD2, EVENTOUT

Table 4.2 shows the pin-out and signal list of the 24-pin FFC connector. Pin 16, 17, 18, 19 of the FFC connector are shared between MCU and Intel® Edison; they form a set of shared interfaces which can be implemented to different functions through a Dual single-pole double-throw (SPDT) Analog Switch. The TS5A23157 Dual SPDT analog switch from Texas Instruments is bidirectional and can handle both digital and analog signals. It contains two sets of SPDT switch, and each group has two switch IOs (NO, and NC), a common terminal (COM) and a switch control signal (IN).

Both of the IN1 and the IN2, the signal select pins of the TS5A23157 are connected to pin PB2 of the STM32F427ZIT7.

When PB2 is LOW, the pin 16 of FFC connector gets connected with pin PF6 of the STM32F427ZIT7; at this time the pin 16 can play a role of SPI5_NCC to form a set of SPI communication interface of MCU along with the pin 17, 18, 19 of the FFC connector.

When PB2 is HIGH, the pin 16 of FFC connector gets connected with pin GP110 of Intel Edison; at this time the pin 16 can play a role of SPI2_FS0 to form a set of SPI communication interface of Intel® Edison along with the pin 17, 18, 19 of the FFC connector. Then Intel® Edison can establish SPI communication with external devices through the FFC connector. In the same time, the pin PF7, PF8, and PF9 of the STM32F427ZIT7 should be set to a high-impedance state.

Additionally, when PB2 is HIGH, the pin PF6 of STM32F427ZIT7 is connected to Intel® Edison's pin GP111 which can feature as SPI2_FS1 to form the SPI communication channel directly between STM32F427ZIT7 and Intel® Edison without extra external wires. At this time the pin 17, 18 and 19 of the FFC connector should leave as not connected. Figure 4.14 illustrates the block diagram of the shared SPI interface.

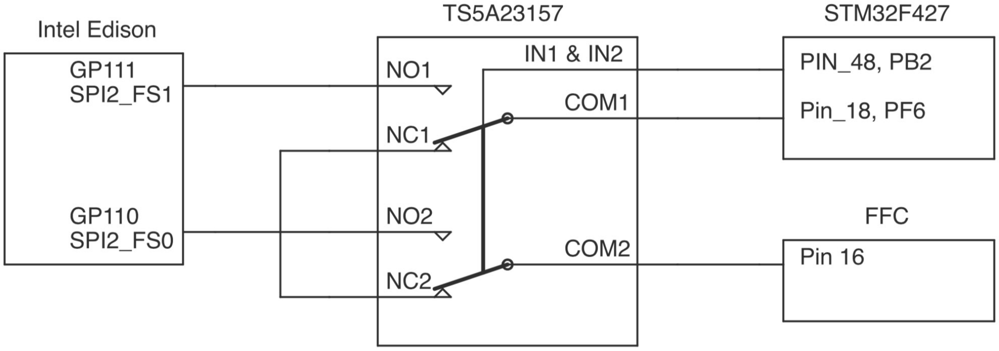


Figure 4.14: Block diagram of shared SPI interface with FFC connector

EM4 uses Panasonic P5KS series narrow pitch connectors for communicating between ioNeurons^{TM} processing modules. A pair of 30-pin connectors are soldered on the top and bottom side of EM4. The header of P5KS connector is on the top, while the socket of the connector is on the bottom. The socket and header are protected from reverse mating by two chamfered corners.

Each connector has a total of 30 pins, which are distributed in two rows with a pin-to-pin pitch of 0.5 mm. It was chosen to be used on EM4 due to the diverse mating height choices for this type of connector. In the design, an 8.0 mm mating height is used to avoid collision between the Intel® Edison module and FM10 when the modules are assembled. Both the header and socket have 4 pins that were defined as power pins: 2 for VCC and 2 for GND; each pin can take up to 0.5A current. This reinforced design effectively avoids the possibility of burning the connector due to large current flow, thereby increasing the capacity to take and supply large current flow. A DB2W900L diode from Panasonic Electronic Components is placed between the power source and power pins to prevent current from going backward. In other words, users cannot power EM4 via power pins from the P5KS connector, but EM4 can provide a 5.0 V power output. Table 4.3 and Table 4.4 show the connection between the 30-pin connectors and the STM32F427ZIT7 with pin's functionalities.

In this thesis, it is defined that any pins on the top header and the bottom socket that have the corresponding position to each other and are connected through PCB via, are Via Pins. A pair of Via Pins carry the same signal. The EM4's P5KS connectors contain 16 pairs of Via Pins: some of them are connected to the MCU's specific pins to build up the I2C or CAN bus communication circuit; while some of them are just pierced forming a bridge without connecting to MCU. For more information about Via Pins, please refer to chapter 2.3.

Table 4.3: EM4 30-pin P5KS Header pin-out and signal list

P5K Pin Number	**P5K Pin Name**	**MCU Pin Name**	**Via Pin**	**Function**
1	HS_1	N/A	Yes	VCC 5V

Continued on next page

Table 4.3 – *Continued from previous page*

P5K Pin Number	**P5K Pin Name**	**MCU Pin Name**	**Via Pin**	**Function**
2	HS_2	N/A	Yes	STM_CAN_LO
3	HS_3	N/A	Yes	VCC 5V
4	HS_4	N/A	Yes	STM_CAN_HI
5	HS_5	N/A	Yes	GND
6	HS_6	N/A	Yes	N_A
7	HS_7	N/A	Yes	GND
8	HS_8	N/A	Yes	N_A
9	HS_9	GP28	Yes	EDISON_I2C_6_SDA, GPIO
10	H_10	N/A	No	N/A
11	HS_11	GP27	Yes	EDISON_I2C_6_SCL, GPIO
12	H_12	N/A	No	N/A
13	HS_13	Edison:GP20, MCU:PB7	Yes	EDISON_I2C_1_SDA, GPIO
14	H_14	GP129	No	EDISON_UART_1_RTS, GPIO
15	HS_15	Edison:GP19, MCU:PB8	Yes	EDISON_I2C_1_SCL, GPIO
16	H_16	GP128	No	EDISON_UART_1_CTS, GPIO
17	HS_17	PB10	Yes	STM_I2C2_SCL, USART3_TX, TIM2_CH3, SPI2_SCK, I2S2_CK, OTG_HS_ULPI_D3, ETH_MII_RX_ER, EVENTOUT
18	H_18	GP130	No	EDISON_UART_1_RX, GPIO
19	HS_19	PB11	Yes	STM_I2C2_SDA, USART3_RX, TIM2_CH4, I2C2_SDA, USART3_RX, OTG_HS_ULPI_D4, ETH_MII_TX_EN, ETH_RMII_TX_EN, EVENTOUT
20	H_20	GP131	No	ESISON_UART_1_TX, GPIO
21	H_21	N/A	No	N/A

Continued on next page

Table 4.3 – *Continued from previous page*

P5K Pin Number	**P5K Pin Name**	**MCU Pin Name**	**Via Pin**	**Function**
22	H_22	N/A	No	N/A
23	HS_23	N/A	Yes	EDISON_PWRBTN
24	H_24	PA7	No	STM_SPI1_MOSI, TIM1_CH1N, TIM3_CH2, TIM8_CH1N, TIM14_CH1, ETH_MII_RX_DV, ETH_RMII_CRS_DV, EVENTOUT, ADC12_IN7
25	HS_25	N/A	Yes	NRST
26	H_26	PA6	No	STM_SPI1_MISO, TIM1_BKIN, TIM3_CH1, TIM8_BKIN, SPI1_MISO, TIM13_CH1, EVENTOUT, ADC12_IN6
27	H_27	PA3	No	STM_UART2_RX, TIM2_CH4, TIM5_CH4, TIM9_CH2, USART2_RX, OTG_HS_ULPI_D0, ETH_MII_COL, EVENTOUT, ADC123_IN3
28	H_28	PA5	No	STM_SPI1_SCK, TIM2_CH1, TIM2_ETR, TIM8_CH1N, OTG_HS_ULPI_CK, EVENTOUT, ADC12_IN5, DAC_OUT2
29	H_29	PA2	No	STM_UART2_TX, TIM2_CH3, TIM5_CH3, TIM9_CH1, ETH_MDIO, EVENTOUT, ADC123_IN2
30	H_30	PA4	No	STM_SPI1_NSS, SPI3_NSS, I2S3_WS, USART2_CK, OTG_HS_SOF, EVENTOUT, ADC12_IN4, DAC_OUT1

Table 4.4: EM4 30-pin P5KS Socket pin-out and signal list

P5K Pin Number	P5K Pin Name	MCU Pin Name	Via Pin	Function
1	HS_1	N/A	Yes	VCC 5V
2	HS_2	N/A	Yes	CAN_LO
3	HS_3	N/A	Yes	VCC 5V
4	HS_4	N/A	Yes	CAN_HI
5	HS_5	N/A	NO	GND
6	HS_6	N/A	Yes	N/A
7	HS_7	N/A	Yes	GND
8	HS_8	N/A	Yes	N/A
9	HS_9	GP28	Yes	EDISON_I2C_6_SDA, GPIO
10	S_10	N/A	No	N/A
11	HS_11	GP27	Yes	EDISON_I2C_6_SCL, GPIO
12	S_12	N/A	No	N/A
13	HS_13	GP20	Yes	EDISON_I2C_1_SDA, GPIO
14	S_14	PC6	No	I2S2_MCK, UART6_TX, TIM3_CH1, TIM8_CH1, EVENTOUT
15	HS_15	GP19	Yes	EDISON_I2C_1_SCL, GPIO
16	S_16	PC7	No	UART6_RX, TIM3_CH2, TIM8_CH2, I2S3_MCK, EVENTOUT
17	HS_17	PB10	Yes	I2C2_SCL, USART3_TX, TIM2_CH3, I2C2_SCL, SPI2_SCK, I2S2_CK, USART3_TX, EVENTOUT
18	S_18	GP135	No	EDISON_UART_2_TX, GPIO
19	HS_19	PB11	Yes	I2C2_SDA, USART3_RX, TIM2_CH4, ETH_RMII_TX_EN, EVENTOUT
20	S_20	GP134	No	EDISON_UART_2_RX, GPIO
21	S_21	N/A	No	N/A
22	S_21	N/A	No	N/A
23	HS_23	PWRBTN	Yes	Power,Sleep button input
24	S_24	PC3	No	SPI2_MOSI, I2S2_SD, EVENTOUT, ADC123_IN13

Continued on next page

Table 4.4 – *Continued from previous page*

P5K Pin Number	**P5K Pin Name**	**MCU Pin Name**	**Via Pin**	**Function**
25	HS_25	N/A	Yes	NRST
26	S_26	PC2	No	SPI2_MISO, I2S2ext_SD, EVENTOUT, ADC123_IN12
27	S_27	PA0	No	UART4_TX, TIM2_CH1, TIM2_ETR, TIM5_CH1, TIM8_ETR, USART2_CTS, EVENTOUT, ADC123_IN0, WKUP
28	S_28	PD3	No	SPI2_SCK, I2S2_CK, USART2_CTS, EVENTOUT
29	S_29	PA1	No	UART4_RX, TIM2_CH2, TIM5_CH2, USART2_RTS, ETH_RMII_REF_CLK, EVENTOUT, ADC123_IN1
30	S_30	PB9	No	SPI2_NSS, TIM4_CH4, TIM11_CH1, I2C1_SDA, I2S2_WS, CAN1_TX, EVENTOUT

4.4 Configuring EM4 with ST-LINK

The ST-LINK can easily programme the STM32F427XX MCU, and it is a debugger and programmer which can communicate with any STM8 and STM32 microcontrollers via the JTAG and Serial Wire Debugging (SWD) interfaces. And the EM4 embedded SWD interfaces. With ST-LINK, the EM4 is fully compatible with ARM® mbed OS which is an open source embedded operating system.

The ST-LINK has a USB connector and 10-pin female connector, in which the USB connector connects to a host computer and the other end connects to the MCU's SWD interface located on the PCB. Developers can use ARM® Keil®, or ARM® mbed online or any GCC-based IDEs to develop and compile codes. The compiled file can be uploaded to MCU through the ST-LINK. Before connecting the EM4 board to

Windows 7, 8 or 10 via the USB connector of ST-LINK, a driver for ST-LINK/V2-1 must be installed.

The SWD interface of EM4 is a DF12(3.0)-10DP-0.5V(86) female connector from Hirose Electric Co Ltd, which has the 10 pins with a small size (5.5mm x 3.8mm). In order to establish the connection between the 10-pin connection header on the ST-LINK and the Hirose connector on EM4, a JTAG adapter was designed. The ST-LINK SWD adapter is the same as the ioNeurons^{TM} FM10 USB-Blaster's JTAG adapter. This SWD Adapter acts as a bridge to connect the ST-LINK and the SWD of EM4. Figure 4.15 shows the signals on the SWD interface of EM4.

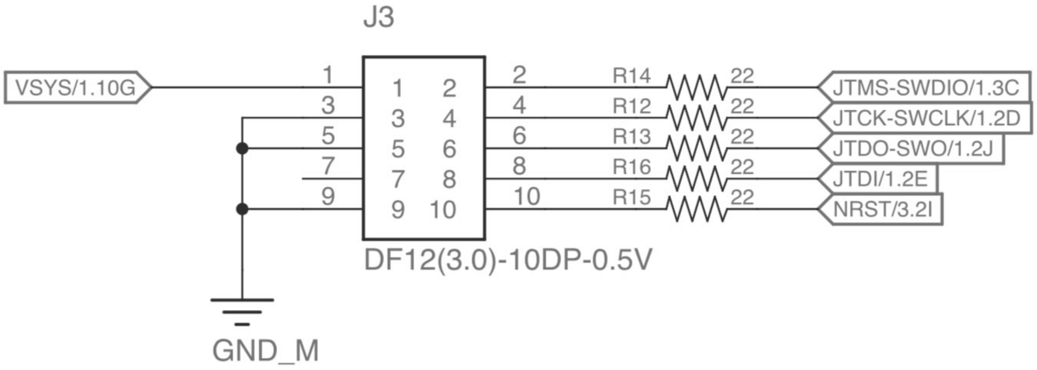


Figure 4.15: The SWD interface schematic in EM4

4.5 Conclusions

The chapter proposes a compact-sized MCU processing module with the stackable interface. Additionally, the EM4 is designed to be compatible with Intel Edison, and the FM10 FPGA module. The study demonstrates 6 layers stack PCB design, power management, connections between SDRAM and MCU chip, onboard external Micro SD card slot, CAN transceiver, and the design of the stackable interface.

The EM4 implements dual data channels using SPDT analog switches, which makes the micro SD card slot shared between MCU and Intel Edison. This design increases the way of data exchanges between processing modules.

The EM4 module can work standalone, it can also collaborate with multiple FM10 modules via the stackable connector. The EM4's microprocessor features an FPU single precision that helps FPGA accomplish complicated floating point calculation avoiding occupying excessive logic gates resources.

Chapter 5

Design and Implementation of the Camera Module

5.1 C5M Board Overview

The C5M is a multi-sensor module for ioNeurons^{TM} FM10. It combines an OV5640 image sensor module, an MPU-9250 motion tracking module and a VL53L0X Time-of-Flight (ToF) ranging and gesture detection sensor all on a small 25x25 mm PCB.

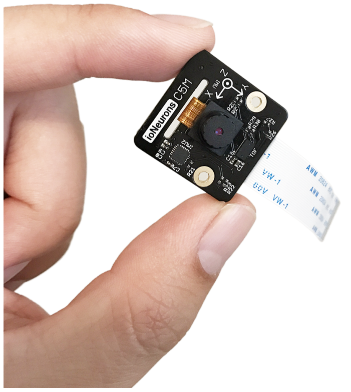


Figure 5.1: ioNeurons^{TM} C5M sensor module

The C5M is a powerful multi-sensor module that provides reliable color image data, motion data, and range data which can be transferred to the ioNeurons^{TM} FM10 FPGA processing module in parallel, based on its 24-pin FFC interface design. The C5M can complete simultaneous sampling of multiple variables in an environment, because of the three different types of sensors. It provides more data to describe the real world in algorithms, thereby allowing implementation of more powerful functions. A recent research-based project from Université Paris-Est implemented an IMU-aided image stacking algorithm in a digital camera for unmanned Aerial vehicles [8].

“The OV5640 image sensor is a low voltage, high-performance, 1/4 inch 5 megapixel CMOS image sensor that provides the full functionality of a single chip 5 megapixel (2592 x 1944) camera using OmniBSI^{TM} technology in a small footprint package. It provides full-frame, sub-sampled, windowed or arbitrarily scaled 8-bit/10-bit images in various formats via the control of the Serial Camera Control Bus (SCCB) interface.” [15].

“MPU-9250 is a multi-chip module (MCM) consisting of two dies integrated into a single QFN package. One die houses the 3-Axis gyroscope and the 3-Axis accelerometer. The other die houses the AK8963 3-Axis magnetometer from Asahi Kasei Microdevices Corporation. Hence, the MPU-9250 is a 9-axis motion tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer and a Digital Motion Processor^{TM} (DMP) all in a small 3x3x1mm package.” [16].

“The VL53L0X is a new generation Time-of-Flight (ToF) laser-ranging module housed in the smallest package on the market today, providing accurate distance measurement whatever the target reflectances unlike conventional technologies. It can measure absolute distances up to 2m, setting a new benchmark in ranging performance levels, opening the door to various new applications. The VL53L0X’s 940nm VCSEL emitter (Vertical Cavity Surface-Emitting Laser), is totally invisible to the human eye, coupled with internal physical infrared filters, it enables longer ranging distance, higher immunity to ambient light and better robustness to cover-glass optical crosstalk.” [17].

5.2 Layout and Components

Figure 5.2 and Figure 5.3 show the front and the back of the C5M PCB layout, and the location of components.

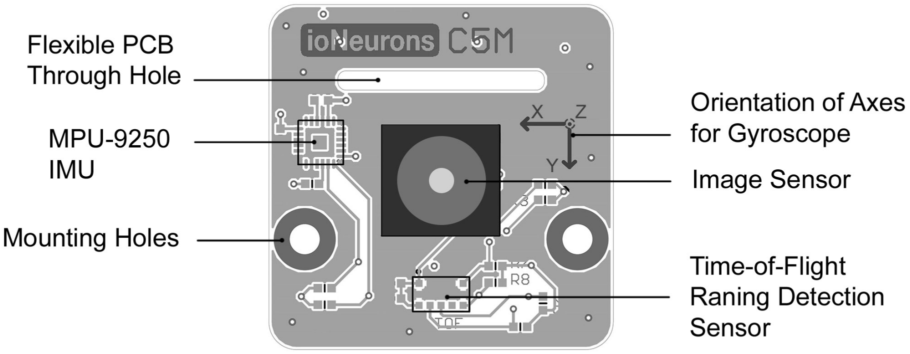

ioNeurons C5M
Flexible PCB Through Hole
MPU-9250 IMU
Mounting Holes
Orientation of Axes for Gyroscope
X
Z
Y
Image Sensor
R8
TOF
Time-of-Flight Raning Detection Sensor

Figure 5.2: The front of the C5M’s PCB and its components

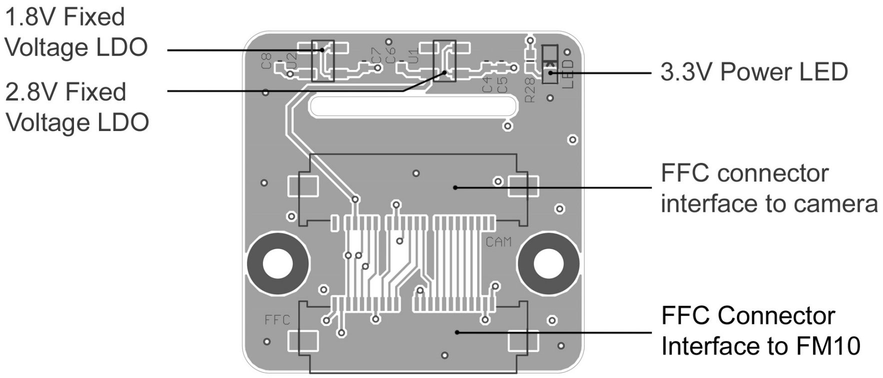


Figure 5.3: The back of the C5M’s PCB and its components

The ToF sensor and the image sensor are placed on the front of the PCB, because they have to face the objects to be measured. The MPU-9250 motion tracking sensor is also integrated on the front of the PCB. The C5M integrates a 1.8 V low dropout (LDO) regulator, a 2.8 V LDO, a power LED indicator and two FFC connectors on the back of the PCB. The upper FFC connector is used to connect the image sensor to the board, and the lower FFC connector is used to connect the C5M to the FM10 FPGA processing module. Both of them are identical to the one used on the FM10, which are 0.5 mm pitch FFC and FPC connectors from Molex. The image sensor is placed in the centre of the PCB. The FPC interface of the image sensor can be inserted into the upper FFC connector through the flat hole located above the image sensor. The power LED indicator is placed on the other side of the PCB to avoid light interference. The diameter of the two mounting holes is 2.2 mm, and a copper sheet, with a width of 2.3 mm around the holes, protects the PCB board. The C5M can be mounted on other equipment using two 2.0 mm diameter screws.

Taking production costs and manufacturing techniques into account,the C5M employs a 4 layer PCB design with plated through holes only; no blind via holes or buried via holes were used since they significantly increase the cost of PCB manufacturing. The minimum diameter of through holes is 0.15mm which is usually the smallest diameter achievable with mechanical drilling. Drilling with a laser can make smaller diameter via holes but this increases the manufacturing cost. The The solder resist layer covers any through holes which are under 0.35 mm in diameter to protect them and avoid a short circuit, and the stop layer mask is not applied to these hole locations. The minimum width of a trace is 0.1524 mm, and the minimum clearance between different signal wires is 0.1778 mm. The minimum distance between objects on signal layers and the board dimension is 0.2032 mm, and the thermal isolation is 0.1778 mm. These parameters are constrained by the production techniques of PCB manufactures.

Table 5.1: EM4 PCB Layer Stack

Layer	**Layer Stack**	**Layer Name**	**Material**	**Thickness**	**Plane**
1		Top paster	Sn63/Pb37	0.12mm	
2		Top overlay			
3		Top solder	Solder resist	0.01mm	
4	- - - - - - - - - - - - -	Top ENIG	Gold	0.05μm	
5	— — — — — —	Top	Copper	0.018mm	Signal
6	///////////////	Dielectric 1	FR-4 TG140	0.0913mm	
7	— — — — — —	Layer 2	Copper	0.03mm	POWER
8	□ □ □ □	Dielectric 2	FR-4 TG140	1.3mm	
9	— — — — — —	Layer 3	Copper	0.03mm	GND
10	///////////////	Dielectric 3	FR-4 TG140	0.0913mm	
11	— — — — — —	Bottom	Copper	0.018mm	Signal
12	- - - - - - - - - - - - -	Bottom ENIG	Gold	0.05μm	
13		Bottom solder	Solder resist	0.01mm	
14		Bottom overlay			
15		Bottom paster	Sn63/Pb37	0.12mm	

FR4 insulation material is applied between any two conductor layers. Table 5.1 shows the PCB layer stack. Electroless Nickel Immersion Gold (ENIG) surface treatment is implemented on the top and bottom layer during PCB manufacture to prevent the nickel from oxidizing with a thin layer of immersion gold.

In order to reduce electromagnetic interference (EMI), the EM4 4-layer structure adopts the top layer, and the bottom layer as signal layers, on which copper is paved in the area around signal routes and connected to the ground plane. The second layer is the power plane, and the third layer is the ground plane. The solid ground plane will minimize inductance, and any signal plane close to the ground plane will have continuous return current. Decoupling capacitors are placed near the power pins of each chip.

5.2.1 Power management and distribution

The C5M module can be directly powered through the FFC connector that integrates a 3.3 V power input interface. When a C5M module is connected to an FM10 module via an FFC, the FM10 provides a stable power source to the C5M module, and then the C5M achieves voltage regulation with two integrated LDO chips. There are three power sails on the C5M: VDD_3.3V, VDD_2.8V, and VDD_1.8V. The VDD_3.3V is system input voltage powering two LDO chips and the MPU-9259 sensor, and it is input from the FCC connector. The VDD_2.8V is used to power the ToF sensor; and the OV5640 image sensor exploits the 2.8 V and 1.8 V inputs to power its I/O interfaces and internal circuit.

When selecting voltage regulators for the C5M, there were two main categories to consider: linear regulators and switching regulators. Linear regulators usually have lower noise, because they use linear, non-switching techniques to regulate the voltage. Compared to linear regulators, switching regulators typically have a wider range of input voltage, higher efficiency, and lower power consumption. A switching regulator can rapidly switch internal FETs elements on and off to store and release energy in order to output the desired voltage. However, the output voltage has more noise than a linear regulator because of the high-frequency switching.

The C5M integrates three different types of sensors. Consideration of the power requirements of these sensors helped inform the choice of voltage regulator. Usually, sensors are sensitive to power supply noise so using a low noise power source for C5M is required to provide high quality, accurate data. In addition to noise, the power consumption also needs to be considered. The OV5640 image sensor needs only 140 mA in active mode; the average power consumption of the ToF sensor is about 20 mW; and the MPU-9250 sensor requires about 3.7mA even with all 9-axes in operating mode. So the OV5640 image sensor and the other sensors are low power devices.

Furthermore, there is not much difference between the input voltage of 3.3V and the output voltages of 2.8V and 1.8V. These two factors combined mean that there will not be significant power consumption in the C5M and therefore, linear regulators are a good choice. The C5M uses LP5907QMFX-1.8Q1 and LP5907QMFX-2.8Q1 which are two linear voltage regulators from Texas Instruments. The former one outputs 1.8 V and the latter outputs 2.8V, and both of them provide ultra-low noise and low quiescent current. Figure 5.4 shows the schematic of 2.8 V LDO in the C5M module.

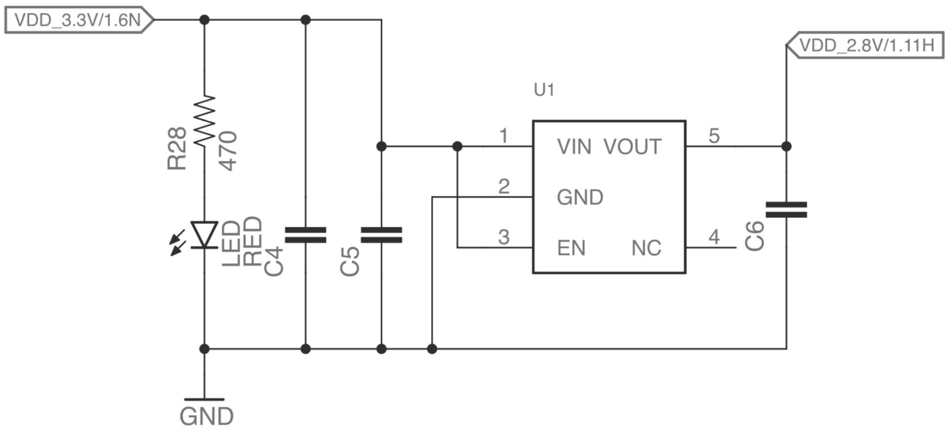


Figure 5.4: The schematic of 2.8 V LDO in C5M

Figure 5.5 shows the schematic of 1.8 V LDO in C5M module.

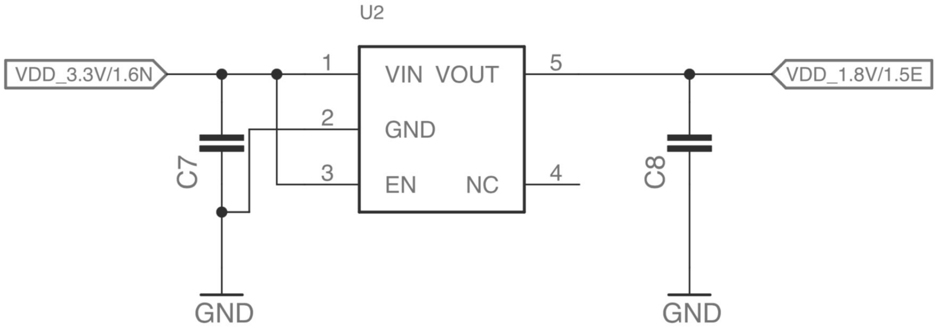


Figure 5.5: The schematic of 1.8 V LDO in C5M

5.2.2 Application schematic

In general, a device that can measure angular rate and acceleration data is called an Inertial Measurement Unit (IMU), and are usually installed in equipment that needs to track and control dynamic motion, such as Unmanned Aerial Vehicles (UAVs), self-balancing scooters and missiles. Typically, the raw data of angular rate provided by a gyroscope cannot be directly used in applications. However, an IMU uses a combination of accelerometers and gyroscopes to calculate the body's specific force and its Euler angles used in the algorithm. When the magnetic field is homogeneous, the data generated from a magnetometer can be used to correct data drift. The MPU-9250 integrates a digital motion processor that implements a simple sensor fusion algorithm to produce a stable orientation. If FPGA or MCU modules are also used, developers can utilize the raw data from all 9 axes to develop and implement more sophisticated sensor fusion algorithms.

The MPU-9250 motion tracking sensor provides standard SPI and I2C communication options for data and command interfaces. For embedded development, with a limited number of FFC connector pins, the C5M module uses an I2C interface that requires only two wires rather than the four wires of an SPI interface. I2C is an open-drain and bi-directional communication standard which is widely used in electronic systems, especially for inter-chip communication.

The MPU-9250 operates as a slave device when connected to an FM10 FPGA module, which acts as the master device. Its slave address is 7 bits long, b110100X, and the last bit is determined by pin AD0 on the chip. When the AD0 is HIGH, the address is b1101001, and when AD0 is LOW, the address is B1101000. This control mechanism allows two MPU-9250 sensors to connect to the same I2C bus. In the C5M module, the AD0 is connected to the ground plane, and its address is locked at b1101000.

The I2C communication standard typically needs two pull-up resistors, which connect the two signal lines and the power supply, to enable communication. When the open-drain interface is not driven, the communication lines are pulled HIGH by the pull-up resistors. Two 2K Ohm pull-up resistors are embedded in the C5M. Figure 5.6 shows the application schematic of the MPU-9250. Document ‘I2C bus pull-up resistor calculations’ [18] indicates the method to calculate resistor values based on different communication speeds requirements.

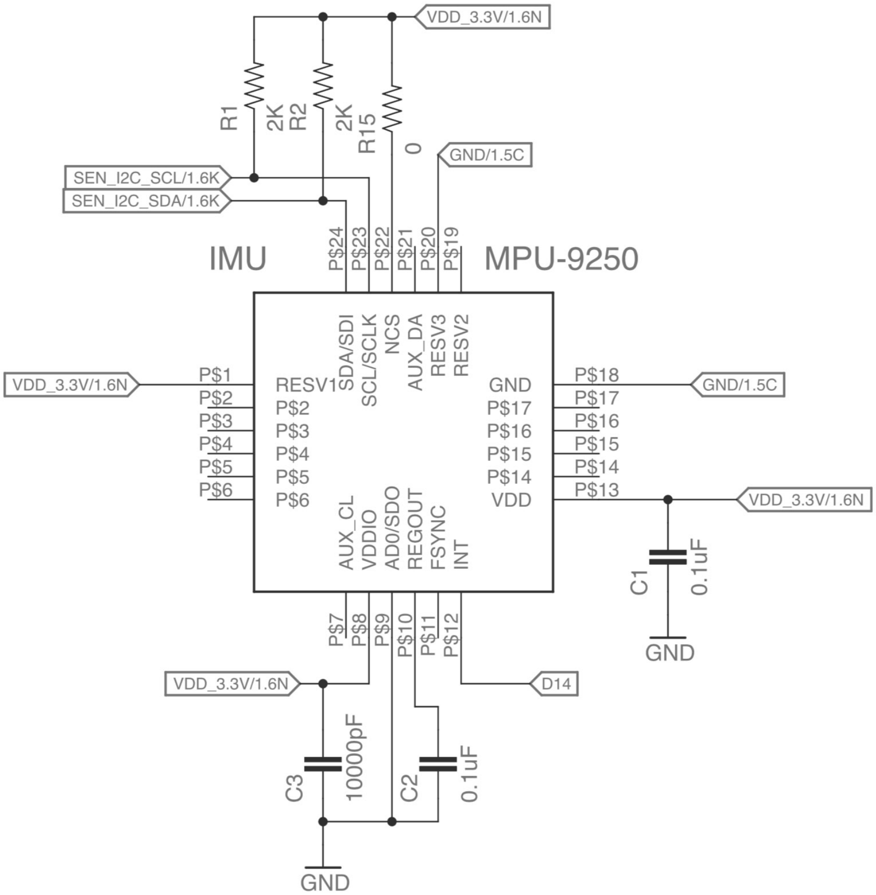


Figure 5.6: The application schematic of MPU-9250 in C5M module

The VL53L0X ToF sensor also provides I2C interface, which is up to 400 KHz. In C5M, it is used as a slave, and its I2C address is 0x52. Figure **??** shows the application schematic of the MPU-9250 in C5M module.

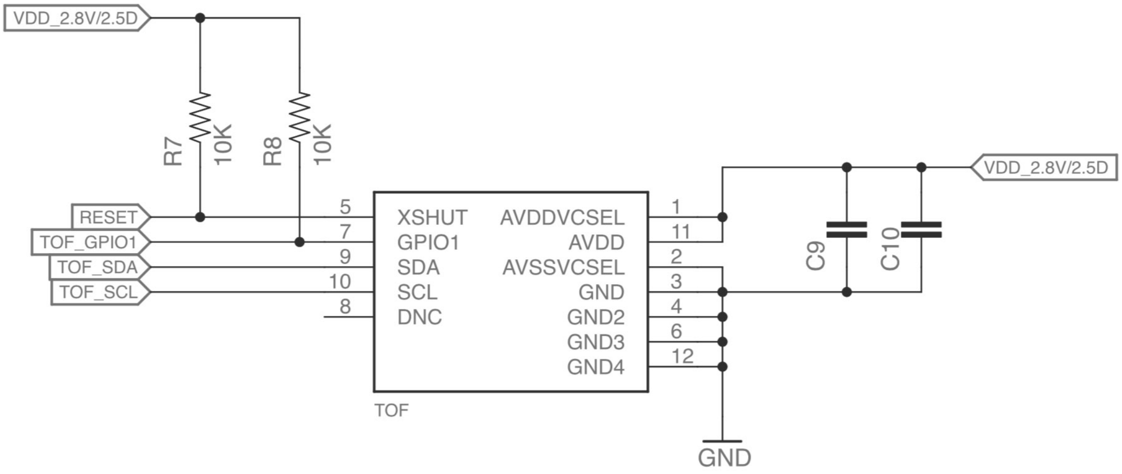


Figure 5.7: The application schematic of VL53L0X in C5M module

5.3 C5M Interfaces

C5M uses 24 conductors 0.5 mm center FFC jumper cable to connect to the FM10 FPGA module.

Table 5.2: C5M 24-pin FFC pin-out and signal list

FFC Pin Number	**Signal Name**	**Type**	**Description**
1	ToF_SDA	Bidirectional	ToF I2C serial data
2	ToF_SCL	Input	ToF I2C serial clock input
3	CAM_SDA	Bidirectional	Camera I2C serial data
4	CAM_SCL	Input	Camera I2C serial clock input
5	CAM_RST	Input	Camera Reset, active low
6	CAM_VSYNC	Output	Camera frame valid, active high
7	ToF_GPIO1	Output	Interrupt output
8	CAM_HREF	Output	Camera line valid, active high
9	CAM_D7	Output	Camera pixel data output D7

Continued on next page

Table 5.2 – *Continued from previous page*

FFC Pin Number	**Pin Name**	**Type**	**Description**
10	CAM_XCLK	Input	Camera input clock
11	GND	Ground	To be connected to main ground
12	GND	Ground	To be connected to main ground
13	VDD_3.3V	Supply	To be connected to 3.3V DC supply
14	VDD_3.3V	Supply	To be connected to 3.3V DC supply
15	CAM_D6	Output	Camera pixel data output D6
16	CAM_D5	Output	Camera pixel data output D5
17	CAM_PCLK	Output	Camera pixel clock output
18	CAM_D4	Output	Camera pixel data output D4
19	CAM_D0	Output	Camera pixel data output D0
20	CAM_D3	Output	Camera pixel data output D3
21	CAM_D1	Output	Camera pixel data output D1
22	CAM_D2	Output	Camera pixel data output D2
23	IMU_SDA	Bidirectional	MPU-9250 I2C serial data
24	IMU_SCL	Input	MPU-9250 I2C serial clock input

5.4 Conclusions

The chapter proposes a multi-sensor module for FM10 and other interface compatible processing boards. The study demonstrates a module design for integrated sensors, which contains an OV5640 image sensor, an MPU-9250 motion tracking module and a

ToF ranging and gesture detection sensor. With this multi-sensor module, the FM10 can get image data, acceleration data and measure angular rate.

Chapter 6

A Small-Scale Quadrotor Helicopter Platform based on ioNeurons Modules

6.1 System Overview

In recent years, multi-rotor drones have developed rapidly. It has brought new opportunities and applications for modern agriculture, industry, military and other fields. Compared with fixed-wing aircraft, drones have the advantages of vertical takeoff and landing, hovering in the air, and flexible operation. In this chapter, we focus on a small-scale Quadrotor Unmanned Aerial Vehicles (UAV) which is developed from the ground up using stackable ioNeurons^{TM} modules. Its simple structure and high reliability provide a high-performance flight platform for research development and education in the field of aerial robotics. Figure 6.1 shows the appearance and the parts of the ioNeurons quadrotor.

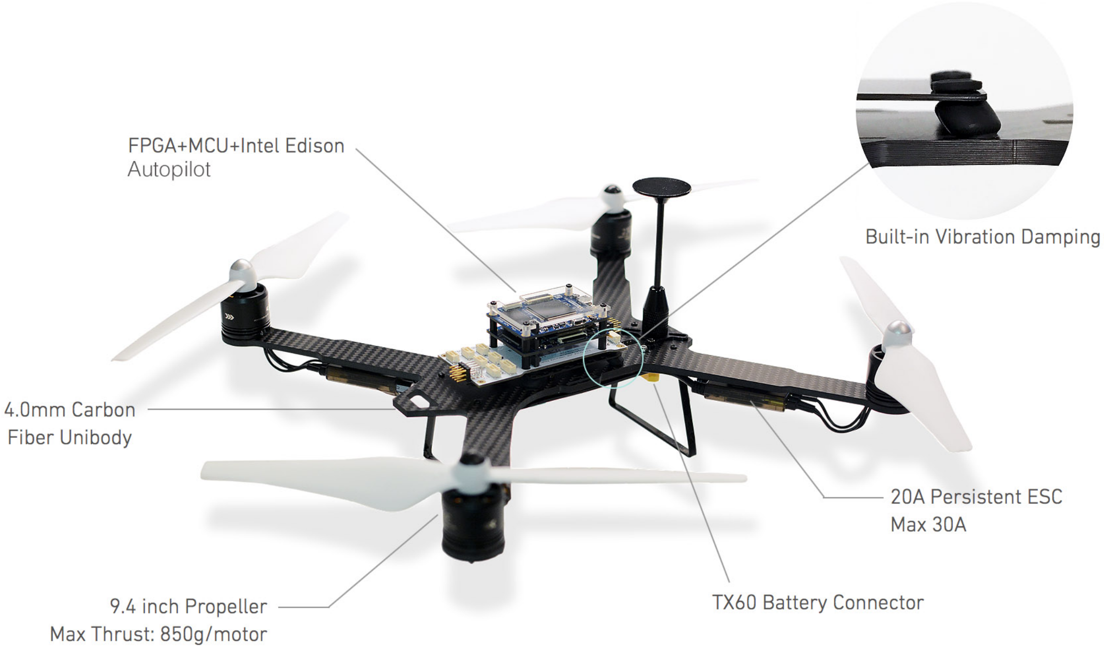


Figure 6.1: The quadrotor built with ioNeurons^{TM} modules

6.2 Design and Construction of a Performance Optimized Quadrotor

The performance of a quadrotor is a combination of results from multiple factors. Flight time is mainly affected by the aircraft weight and battery capacity. Flight stability and automatic flight modes are primarily affected by its flight controller. In order to design and construct a performance-optimized quadrotor, it is necessary to comprehensively consider the components that make up the drone. The main components of a quadrotor include a frame, a propulsion system, batteries, a flight controller, a wireless data transceiver, and external sensors such as GPS and camera. The propulsion system contains motors, electronic speed controllers (ESCs) and propellers. Each component has an affect on the flight performance of the drone.

The ioNeurons quadrotor is packed with serious processing and computing power, deploying an advanced FPGA, MCU and on-board computing module on a single platform. Interfacing the on-board high-performance sensors and actuators or additional devices is a breeze with the flexibility of the numerous communication options provided. From low-level control algorithms to high-level communications, this flight platform is fully programmable to customize and tailor the autopilot. Figure 6.2 shows the block diagram of the quadrotor.

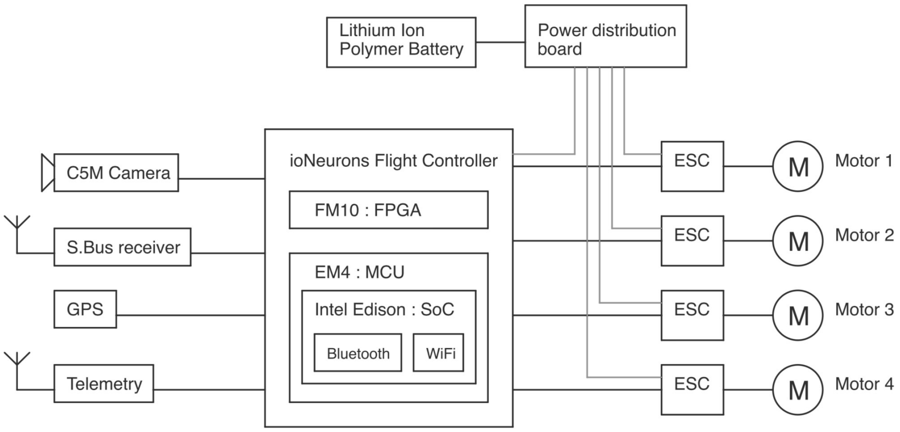


Figure 6.2: Block diagram of the ioNeruons quadrotor helicopter

6.2.1 Unibody frame

When designing a quadrotor helicopter, typically the first thing to do is to determine its application direction. The ioNeurons quadrotor is designed for education and research development, so a small body size and flexible movement are the preferred design philosophy considered throughout the entire design process. The classic quadrotor frame is a symmetrical cross shape, and consists of four arms which are symmetrically arranged in front, rear, left and right direction around the center body. Each arm has its own motor and propeller at the same height. The basic structure of a quadrotor's frame is shown in Figure 6.3.

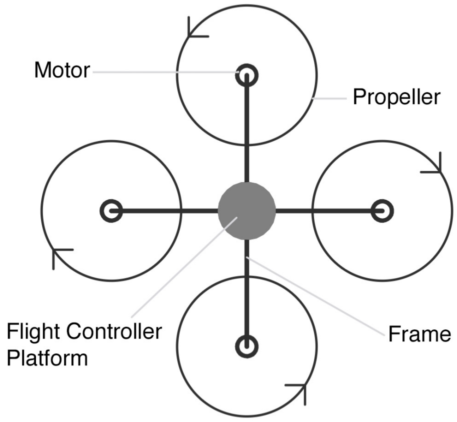


Figure 6.3: Basic structure of a quadrotor's frame

Concerning the typical dimensions of motors and propellers on the market, the ioNeurons quadrotor is designed to have a wheelbase of 370 mm and an arm length of 140 mm. The length of the arm should not be less than half the length of the propeller. Otherwise, the propeller will pass through the top of the frame center area when it rotates, and the frame body will block the generated air flow, causing air spoil, vibration, and noise. There should be no obstructions below the propellers, as this will increase the effective power of propellers.

A high-performance frame should take mechanical properties, materials, and cost-effectiveness into consideration. The ioNeurons quadrotor frame is made of carbon fiber reinforced polymer (CFRP). This material is remarkably strong, with excellent stiffness, and light weight. Therefore, it is widely used in aerospace and automotive applications. The entire frame of the quadrotor is cut from a single piece of CFRP. The seamless connections between arms and body avoid the use of screws. This kind of structure not only increases the strength of the frame but also reduces the weight of the drone, while avoiding the safety issues of loosening screws. Using a single piece of CFRP for the frame can be referred to as a unibody frame. It is responsible for carrying all electronic devices and batteries mounted either on the top or bottom of the frame. The unibody frame also bridges the four motor thrusts to overcome the payload. The thickness of the unibody frame is 3.5 mm which provides excellent load carrying capacity, while its weight is only 135 g.

The unibody frame also packs a built-in vibration damping structure which provides a small platform to place the flight controller. This small platform is connected to the unibody frame by four soft cushion rubber pads which make a non-rigid connection that significantly reduces body vibration transmitted to the flight controller, and offers the IMU a quiet place to avoid the interference of vibration. The unibody frame is a prerequisite for the quadrotors' good performance.

6.2.2 Propulsion System

The propulsion system provides motive power for flights. An excellent propulsion system must be reliable and efficient. It usually includes a three-phase brushless DC (BLDC) motor, an Electronic Speed Controller (ESC), and a propeller. A brushless DC motor is a synchronous motor consisting of a rotor and a stator. It has a simple structure and does not use carbon brushes, slip rings or other mechanical components, which significantly improve its service life and reliability. At the same time, it has the advantage of high operating efficiency and good speed control performance. Brushless DC motors have been widely used in the field of drones.

According to the principle of the brushless DC motor operation, it must use a device that can control the current flowing within the stator windings so that a torque interaction between the rotor and the stator can be kept constant and the energy conversion of the motor can reach an optimal state. Such a device is called an inverter. In the field of drones, the inverter is also called an ESC. Without an ESC, the brushless DC motor cannot work. In general, an ESC's performance is distinguished based on its operating voltage range and the maximum drive current that can be provided. ESCs which have high-efficiency drive capabilities are a guarantee of an efficient propulsion system and is a prerequisite for more extended flight time.

ESCs can be divided into sensored and sensorless. The main difference is the strategy they use to detect the position of the BLDC motor's rotor. Sensored ESCs typically use Hall sensors which are placed around the motor in the right position to detect the rotor position for the current control. This type of ESC has excellent performance in both low-speed and high-speed operations. Moreover, the BLDC motor rotating speed can be adjusted quickly and precisely. However, the installation of Hall sensors is relatively complicated. Sensorless ESCs estimate the position of the rotor by detecting the back EMF. For this reason, a sensorless ESC does not work well for low-speed control. For a high-speed BLDC motor, this technique also requires a high-performance microprocessor to handle and perform position estimation of the rotor.

The efficiency of a propulsion system is the result of the interaction between the motor and its propeller. If the motor velocity constant and the size of the propeller cannot be matched, the flight time will be reduced. In severe cases, the drive current will be too large, and the ESCs will burn up. Based on comparisons with existing products on the market, the ioNeurons quadrotor uses DJI's E305 tuned propulsion system, which contains a 2313E BLDC motor, an ESC with a maximum drive current of 30A, and a model 9450 propeller. The motor weighs 56 g, and its stator size is 23 mm x 12 mm. The ESC uses a 3 or 4 cell Lipo battery as the power supply. The length of the propeller is 24 cm which corresponds to the size of the unibody frame. With this propeller, the maximum thrust of the motor can reach 850 g. It is very suitable for an aircraft with a takeoff weight of 1 ~ 2.5 kg.

The flight controller weights 58 g, and the unibody frame weights 135g. The three cell 1550 mAH Lipo battery from Genace & Tattu weights 140g. So the total weight of the ioNeurons quadrotor is 690 g. Without the battery, it only weighs 550g. The takeoff weight of the quadrotor can be 2.5 kg so that it can lift 1.8 Kg of payload; this is almost three times its own weight.

6.2.3 Design of the ioNeurons Autopilot platform based on FM10, EM4 and C5M

The flight controller of the ioNeurons quadrotor consists of an FM10 FPGA processing module, an EM4 microcontroller processing module, and an interface & sensor board designed explicitly for the quadrotor. Without changing any FM10 and EM4 design, a specific application platform is easily and simply built. And this platform provides powerful real-time computing capabilities to developers. The flight controller takes advantage of the FPGA's parallelism and environment for reliable hard implementation of programs. At the same time, the EM4 features a single-precision floating point unit that can help the FPGA accomplish complicated floating point calculations and be in charge of middleware executions. The Intel® Edison computing module can connect with the EM4, enabling a Linux operating system on a compactly sized flight controller. Figure 6.4 shows the flight controller, as shown from top to bottom, they are the EM4, the FM10, and the interface & sensor board.

The flight controller runs a multitasking program. In order to ensure the stability of the quadrotor flight, the requirements for real-time calculations are stringent. To calculate the speeds of all motors, the flight controller must read various data from multiple sensors, while also maintaining communication with the ground control station. If a controller offers visual aided flight functions, it also needs to execute image processing algorithms at the same time, which undoubtedly aggravates the calculation tasks of the flight controller. If a single high-performance microcontroller handles all the tasks, the real-time performance of the calculation becomes worse and worse

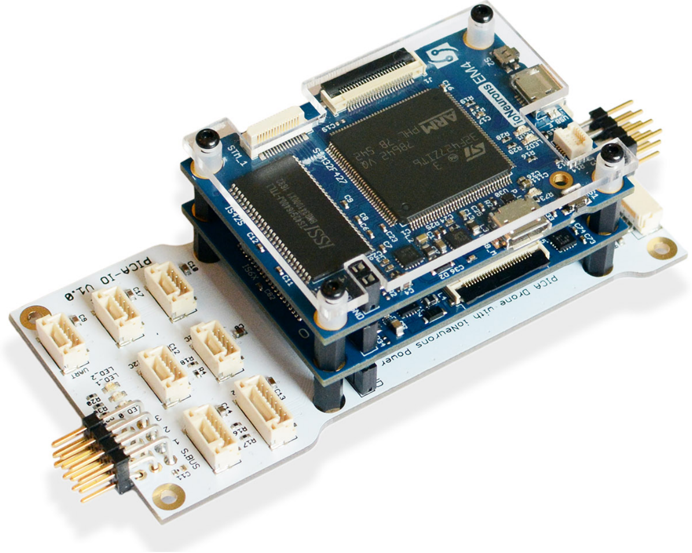


Figure 6.4: ioNeurons^{TM} Autopilot

as the number of sensors and functions increases. This can eventually lead to flight instability and even the quadrotor getting out of control. The design of the ioNeuorns flight controller provides great help to solve the above issues. It can classify tasks and operations based on the different requirements of the real-time calculations, and ultimately execute different levels of tasks on different processing modules. Figure 6.5 illustrates the tasks with different real-time requirements.

The FPGA should be used to implement algorithms which need the most stringent real-time requirements, such as the low-level stabilization control. In this flight controller, it should read the data of sensors, which is directly related to flight stability, and complete the stabilization algorithm to control the speed of each motor directly to achieve stabilized autonomous flight. The frequency of these low level calculations should be around 400 Hz or above. The FPGA can also perform image processing at the same time. Due to the FPGA's parallel operation, image processing does not affect the execution of other algorithms. With this flight controller, the FM10 FPGA module can connect with the C5M camera module and access the IMU and ToF

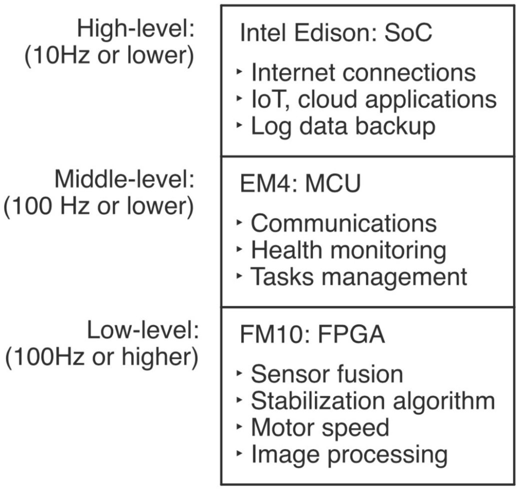


Figure 6.5: Tasks with different real-time requirements

sensors integrated on the C5M to achieve the dual IMU flight controller.

The MCU is used to perform tasks which have relatively low real-time requirements, such as a middle-level controller. The FPGA module is in charge of low-level functions which frees up the MCU and gives it more time and resources to handle its related tasks. Compared with the FPGA, the software for the MCU has much better portability of design; it also typically has shorter development iterations. The MCU can manage communications between the flight controller and the ground control station, parse the command packets and forward to the FPGA for execution. It also can be used for collecting the FPGA feedback information, monitoring system health, and generating and storing log files. The MCU provides communication interfaces with other external devices. Most of these tasks are performed at 50 to 200 Hz.

Intel® Edison provides developers with a computing module on which users can run a small Linux system. The inclusion of an operating system, allows the implementation of high-level controllers which are responsible for tasks that are not sensitive to real-time performance. Such tasks will not affect flight stability even if there is a delay in transmission or execution. These controllers are the highest level ones in the application, and they can perform tasks that are usually difficult to implement on FPGAs or MCUs. For example, they can to manage Internet connections or implement Internet of Things (IoT) applications; they can also upload the MCU's log files to the cloud, or download missions from the cloud servers for remote control. Most of these tasks are usually performed at 10Hz or even lower.

6.2.3.1 Design of the inter-component communication

The flight controller benefited from the ioNeurons^{TM} stackable interface design. It easily supports and implements multiple communication standards which not only serve inter-component communications but also establish communications with external devices. The flight controller uses three different types of computing chips; they are FPGA, microcontroller, and SoC. Establishing multiple communication methods between them is a way to ensure that data can be exchanged at high speed. The current design allows direct communication between any two computing chips with more than one communication standards. A block diagram is shown in Figure 6.6 that depicts the connections and communication protocols used for the flight controller.

The stackable interface connector contains three I2C interfaces (I2C_1, I2C_2, and I2C_3) and three UART interfaces; they are separated and do not affect each other. This design enables any two modules to have their own independent I2C channels and UART channels. In addition to this, the faster SPI communication protocol can be used between FPGA and MCU, and MCU and Intel® Edison. The USB interface of the FM10 and the USB OTG interface of the EM4 are not shown in the figure, both of them can be used to connect to a computer to debug the flight control software. The rich internal communication channels enable unobstructed data exchanges between different modules so that each module can be fully developed.

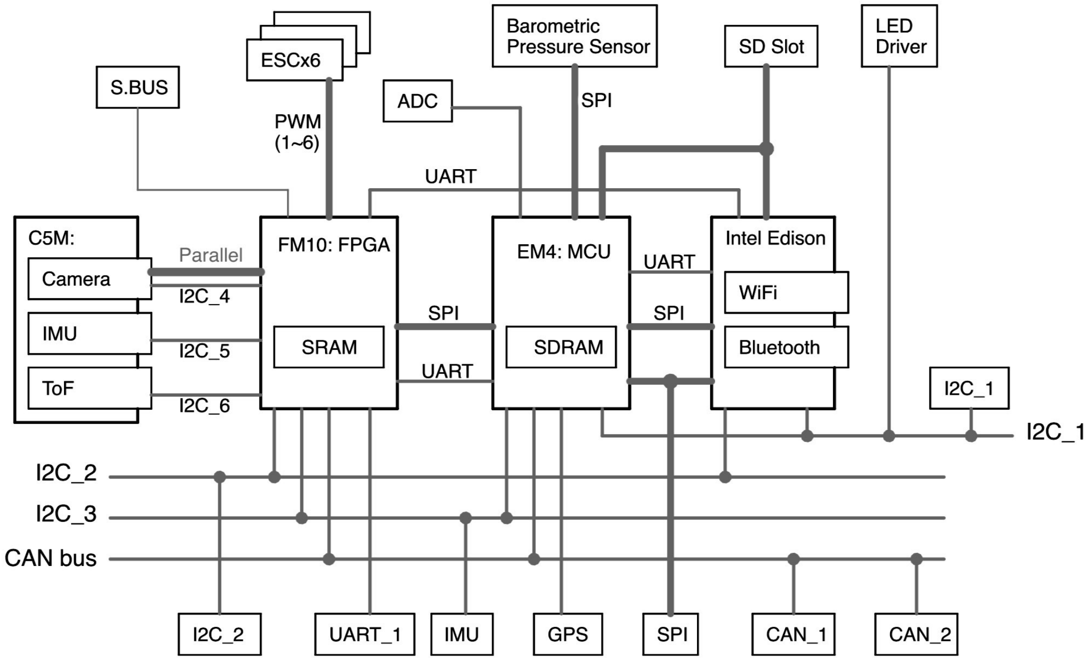


Figure 6.6: The block diagram depicts connections and communication protocols of the flight controller

6.2.3.2 Design of the interface board for ioNeurons processing modules

As a bridge, the interface board connects external devices and the ioNeurons^{TM} processing modules. It also provides developers with communication interfaces for accessing modules, and power interfaces. The processing modules sit on the top of the interface board and ensure the reliability of the interface connection through four mounting holes. A variety of communication connectors are integrated on both sides of the flight control board, including CAN, SPI, I2C, and UART, and other communication standard interfaces for connecting GPS, telemetry, and other sensors. These interfaces also contain power and ground lines which can power devices directly. In addition, there is a connector which has two ADC channels that users can use to connect sensors whose output signals are analog signals. These connectors are JST's GH connector which offers a secure locking structure and contact lock mechanism. These features ensure the reliability of the connection without being easily detached by external forces. Figure 6.7 illustrates the appearance and connectors of the interface board. Table 6.1 lists all interfaces and connections between sensors and processors.

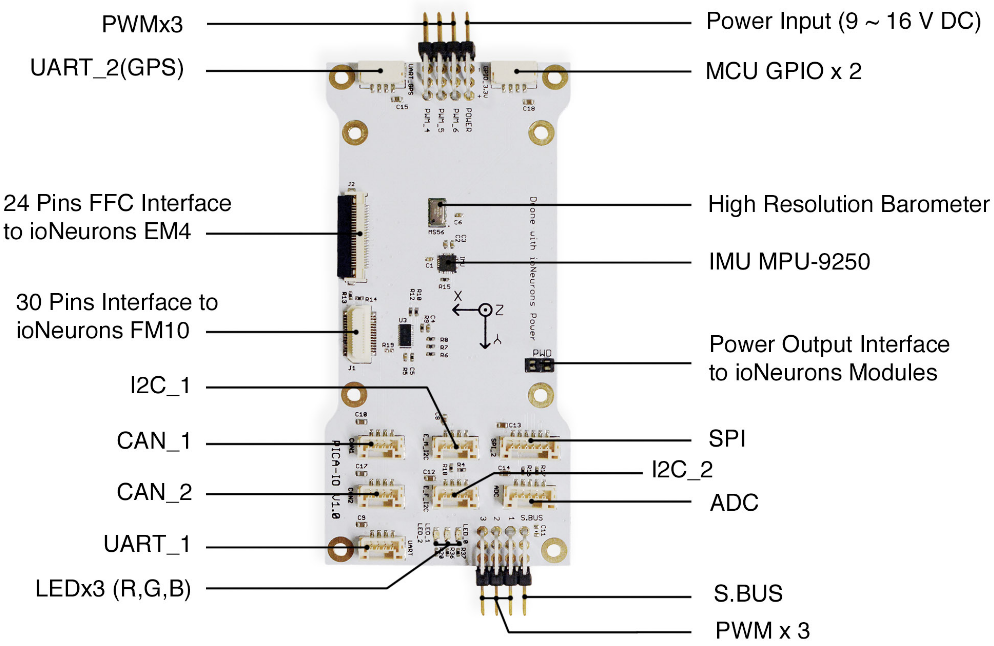

PWMx3
Power Input (9 ~ 16 V DC)
UART_2(GPS)
MCU GPIO x 2
24 Pins FFC Interface to ioNeurons EM4
High Resolution Barometer
IMU MPU-9250
30 Pins Interface to ioNeurons FM10
X
Z
Y
Power Output Interface to ioNeurons Modules
I2C_1
CAN_1
SPI
I2C_2
CAN_2
ADC
UART_1
LEDx3 (R,G,B)
S.BUS
PWM x 3

Figure 6.7: ioNeurons$^{\mathrm{TM}}$ quadrotor interface board

The interface board can output up to six Pulse Width Modulation (PWM) signals to control the motor speed. According to the needs of developers, it not only supports the development of a Quadcopter but can also be used to develop a Hexacopter flight controller. A protection circuit was added between the PWM outputs and the FPGA pins to isolate the low voltage control circuit from the high voltage circuit. This provides ESD protection and prevents the high voltage that may be generated when an ESC is burned from passing to the FPGA pins and causing the master controller to be damaged.

The interface board integrates two additional sensors. The first is the IMU MPU-9250 which is placed in the center of the area where the processing module is installed; this position is more conducive to obtaining an accurate flight attitude. The second sensor is a high resolution barometric pressure sensor, and its model number is MS5611-01BA03 from TE Connectivity Sensor Solutions. It offers a 10 cm resolution, and allows users to obtain the current flying height based on air pressure. A sponge is installed to cover the sensor to avoid windy disturbances. In addition to these sensors, RGB tri-color LEDs are integrated on both sides of the PCB, enabling the flight controller to indicate the current working status intuitively through different LED colors.

Table 6.1: The pinouts of connectors and sensors on the interface board.

Connector Name	**Pin Num**	**Signal Name:FPGA Pin Number**	**Signal Name:MCU Pin Name**	**Signal Name: Edison Pin Name**	**Power**
CAN_1	1				5V
	2	CAN_H, (RX:PIN_132)	CAN_H, (RX:PB12)		
	3	CNA_L, (TX:PIN_131)	CAN_L, (TX:PB13)		
	4				GND
CAN_2	1				5V
	2	CAN_H, (RX:PIN_132)	CAN_H, (RX:PB12)		
	3	CNA_L, (TX:PIN_131)	CAN_L, (TX:PB13)		
	4				GND
UART_1	1				5V
	2	TX:PIN_13			
	3	RX:PIN_12			
	4				GND
UART_2	1				5V
	2		TX:PG14		
	3		RX:PG9		
	4				GND
I2C_1	1				5V

Continued on next page

Table 6.1 – *Continued from previous page*

Connector Name	**Pin Num**	**Signal Name:FPGA Pin Number**	**Signal Name:MCU Pin Name**	**Signal Name: Edison Pin Name**	**Power**
	2		SCL:PB8	SCL:GP19	
	3		SDA:PB7	SDA:GP20	
	4				GND
I2C_2	1				5V
	2	SCL:PIN_140		SCL:GP27	
	3	SDA:PIN_135		SDA:GP28	
	4				GND
SPI	1				5V
	2		NSS:PF6	NSS:GP100	
	3		SCK:PF7	SCK:GP109	
	4		MISO:PF8	MISO:GP114	
	5		MOSI:PF9	MOSI:GP115	
	6				GND
ADC	1				5V
	2		ADC1:PC1		
	3				GND
	4		ADC2:PC4		
	5				GND
PWM1	1	PIN_8			
PWM2	1	PIN_141			
PWM3	1	PIN_123			
PWM4	1	PIN_11			
PWM5	1	PIN_14			
PWM6	1	PIN_10			
S.BUS	1	PIN_7			
I2C_1	1				5V
	2		GPIO1:PB15		
	3		GPIO2:PB1		
	4				GND
Barometer I2C			SCK:PE2		
			MOSI:PE6		
			MISO:PE5		
			NSS:PG13		

Continued on next page

Table 6.1 – *Continued from previous page*

Connector Name	**Pin Num**	**Signal Name:FPGA Pin Number**	**Signal Name:MCU Pin Name**	**Signal Name: Edison Pin Name**	**Power**
IMU I2C		SCL:PIN__25	SCL:PB10		
		SDA:PIN__24	SDA:PB11		
LED Driver			SCL:PB8	SCL:GP19	
			SDA:PB7	SDA:GP20	

6.3 Results and Conclusion

The small-scale quadrotor drone built using ioNeurons^{TM} modules provides an optimized flight platform for research development and education in the field of aerial robotics. The ioNeurons^{TM} modules shorten the time of flight controller hardware design, and also provide a rich interface and powerful computing capabilities for software development. The flight controller takes the advantage of FPGAs, MCUs, and SoCs. Developers can implement complex multitasking programs that can run on different processing modules depending on the task's requirements for real-time performance. The small-scale quadrotor also uses a customized unibody frame which reduces the weight of the frame while improving the strength.

Pixhawk 1 is an open source flight controller, which is made by the PX4 team at ETH Zurich. The Following list shows the comparison of two flight controllers.

Table 6.2: Comparison between Pixhawk 1 and ioNeurons flight controller.

Features	**Pixhawk**	**ioNeurons**
Support customized and tailored	Yes	Yes

Continued on next page

Table 6.2 – *Continued from previous page*

Features	**Pixhawk**	**ioNeurons**
Support PX4 software	Yes	Yes
Size(w x l)	50mm x81.5mm	45mm x 85mm
Weight	38g	58g
MCU	STM32F427	STM32F429
SDRAM	No	64MB
SRAM	No	4MB
FPGA	No	Intel Max 10
Camera	No	ioNeurons C5M
IMU	ST Micro L3GD20	Invensense MPU9250
PWM	6	6
Input voltage	5V DC	5 16V DC
I2C	Yes	Yes
CAN	Yes	Yes
SPI	Yes	Yes
microSD card	Yes	Yes
SoC	No	Intel Edison
WiFi	No	Yes
Bluetooth	No	Yes

This light weight quadrotor has powerful thrust and computing capabilities as a performance optimized flight platform which meets size, weight, and power (SWaP) requirements.

Chapter 7

Conclusions

This thesis presents a stackable development platform which adopts a modular design method. Developers can mix and match a module's individual capabilities according to the project's needs, making every solution a custom tailored solution without the headache of a custom built platform. In this thesis, a quadrotor flight controller is built with the ioNeurons$^{\text{TM}}$ modules. It shows the module's powerful computing capability and flexible custom function interfaces.

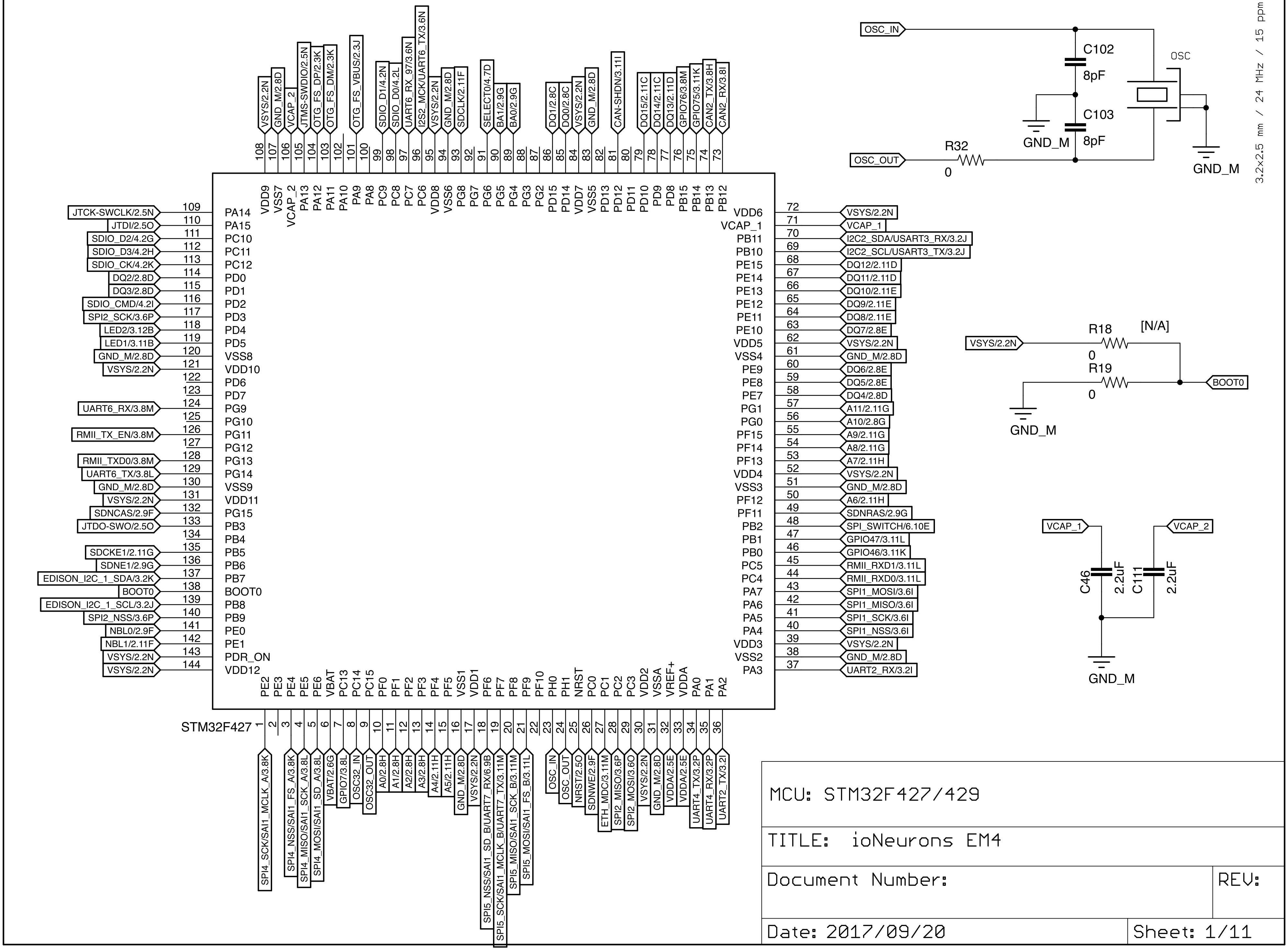
According to the different functions of these modules, they are divided into processing modules, sensor modules, and interface modules. These modules constitute a hardware platform which has extensible capabilities to provide high-speed, parallel computing power with low power consumption. Based on the particular stackable interface design, it can integrate FPGA, MCU, and SoC modules into one hardware platform. Developers can assign tasks to multiple processing modules based on the different real-time requirements of calculations. This thesis introduces the implementation of a scalable development platform for complex real-time systems.

For a complex system, developers usually specify the interface between modules, then assign related tasks to different modules. The modular design presented here allows easy division of multi tasks and isolation of tasks so that team collaboration can be easily achieved. It is the opinion of the author that method of modular design will continue to be developed in the future for many years. In the future, dozens or even hundreds of processing units may replace a powerful processor in applications. Each processing unit completes similar but slightly different operations, each of them performs simple tasks but together they form a complex system.

Similar theories have been implemented in other field. Today hundreds of millions of transistors are put together to make a powerful CPU. But, in 1971, Intel introduced the world's first microprocessor 4004 which contained only 2300 transistors. In the not-too-distant future, when Moore's Law no longer holds, people may seek new directions in hardware structures and computing architecture. There will be a variety of simple computing modules in a system, and they will be connected to each other in large numbers to complete a complicated calculation. The word "neurons" is used to name the modules, which is a good wish for the future. Each module is connected like a neuron in the human brain and they combine to form a complex computing architecture.

Appendix A

Schematic



MCU: STM32F427/429

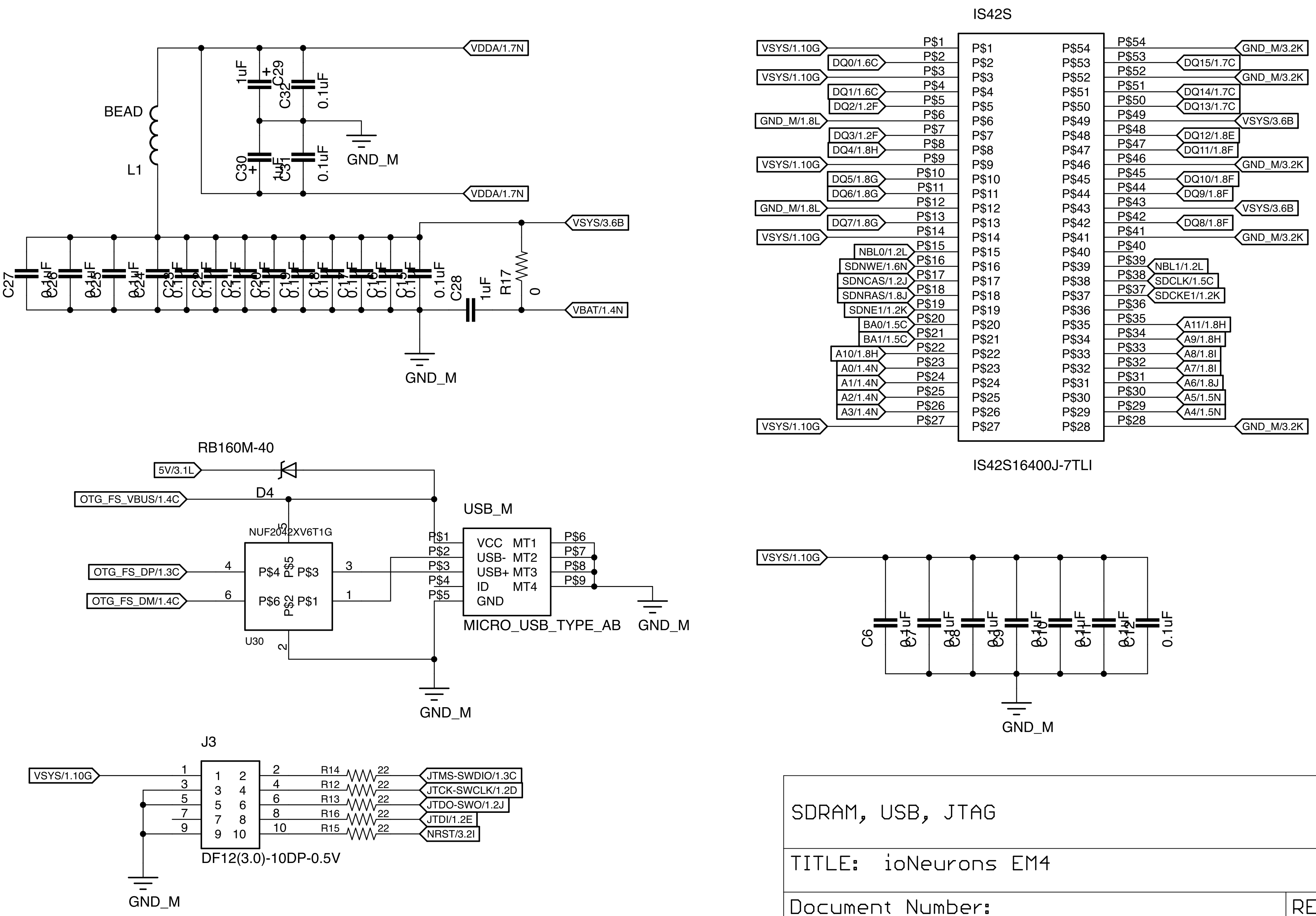
TITLE: ioNeurons EM4

Document Number:

REV:

Date: 2017/09/20

Sheet: 1/11

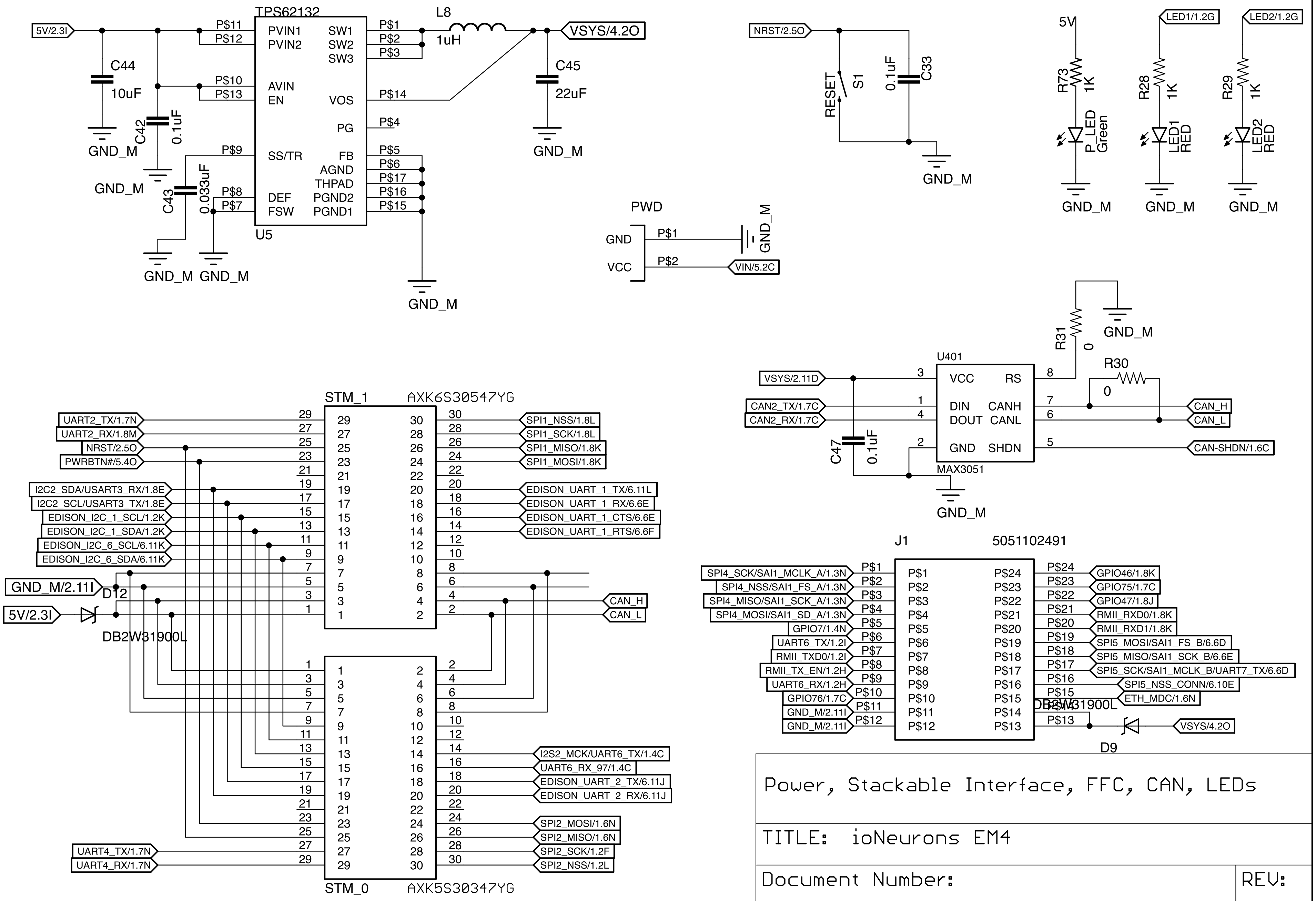


SDRAM, USB, JTAG

TITLE: ioNeurons EM4

Document Number: REV:

Date: 2017/09/20 Sheet: 2/11



Power, Stackable Interface, FFC, CAN, LEDs

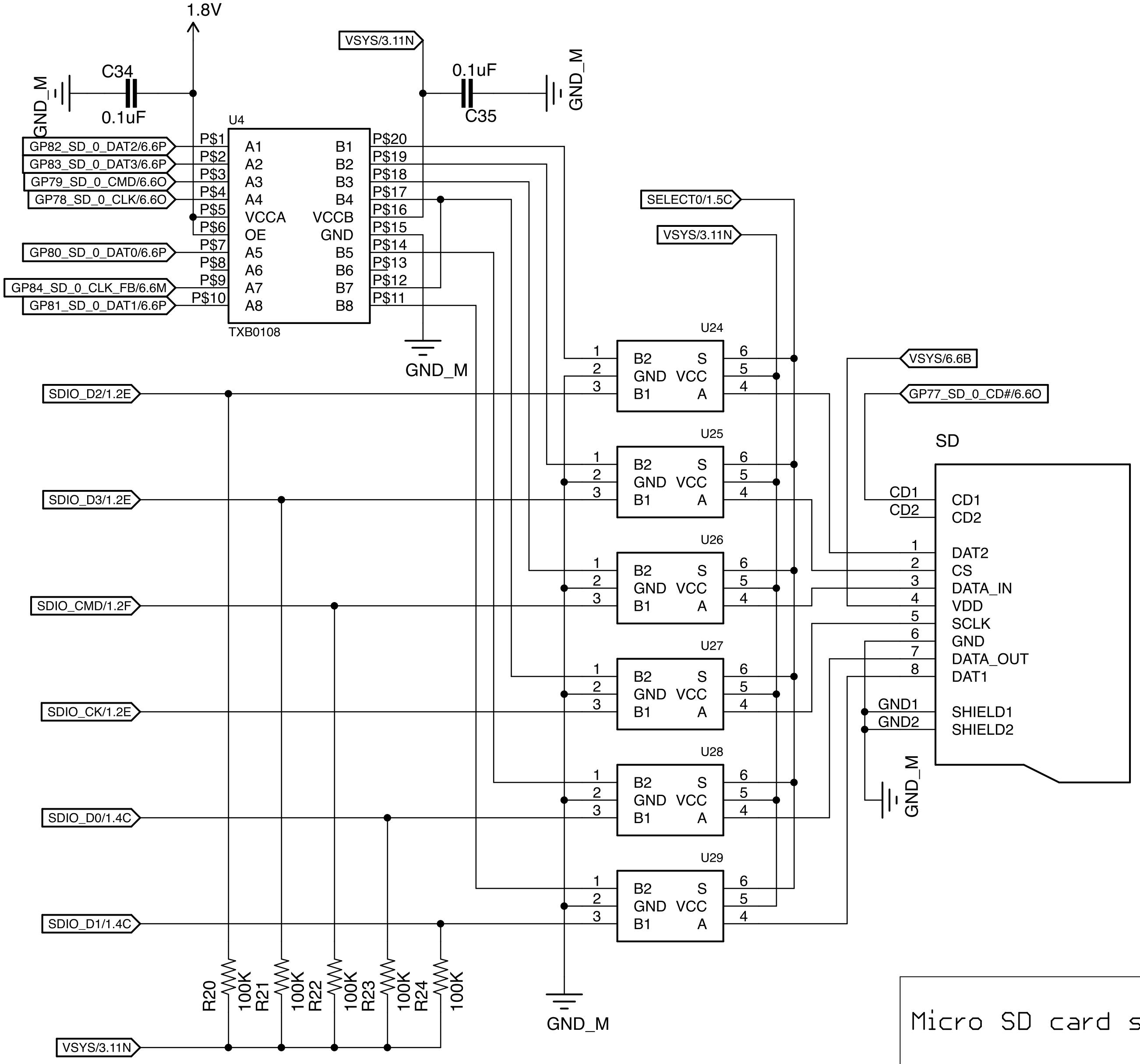
TITLE: ioNeurons EM4

Document Number:

REV:

Date: 2017/09/20

Sheet: 3/11

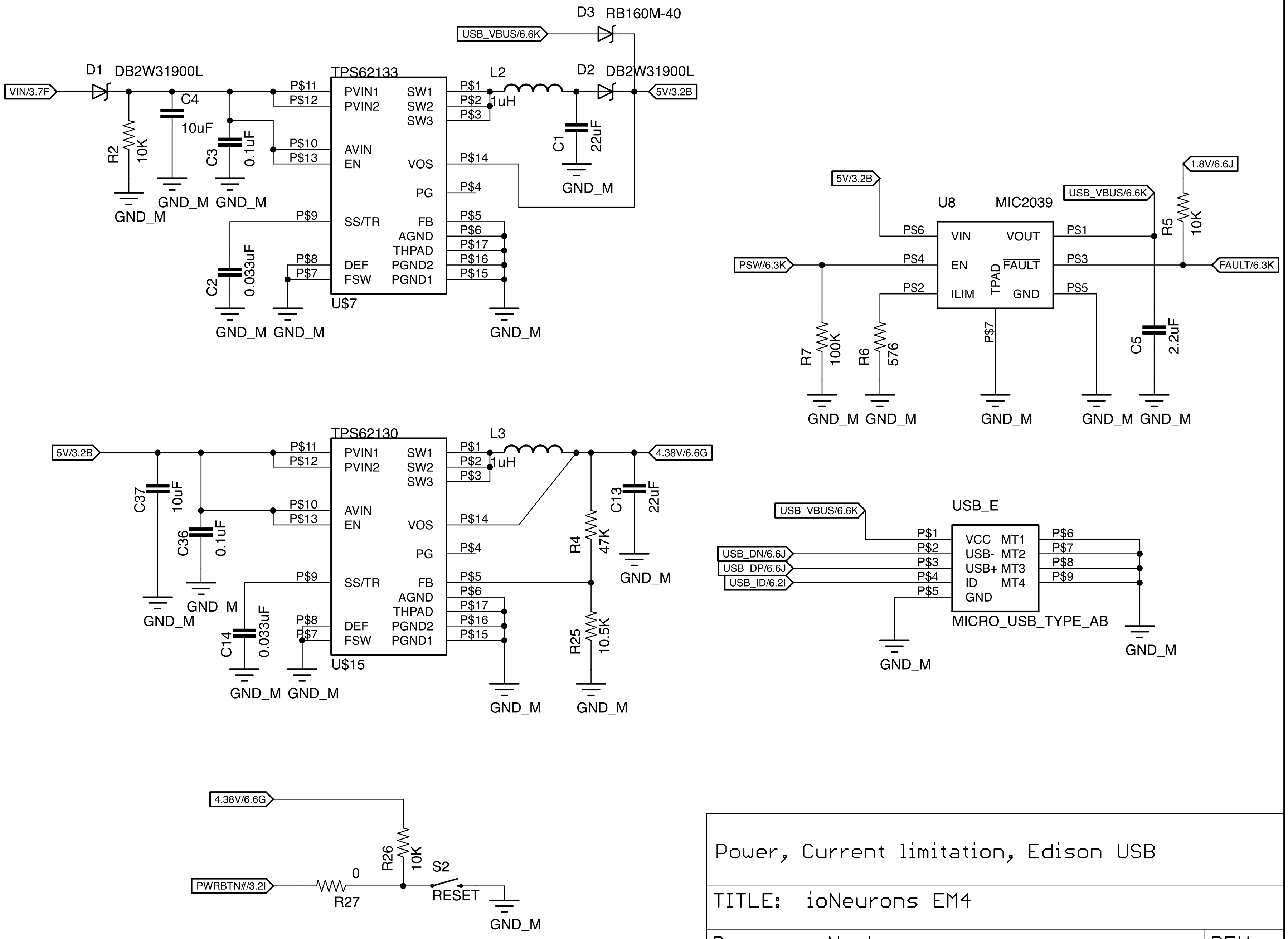


Micro SD card slot

TITLE: ioNeurons EM4

Document Number: REV:

Date: 2017/09/20 Sheet: 4/11



Power, Current limitation, Edison USB

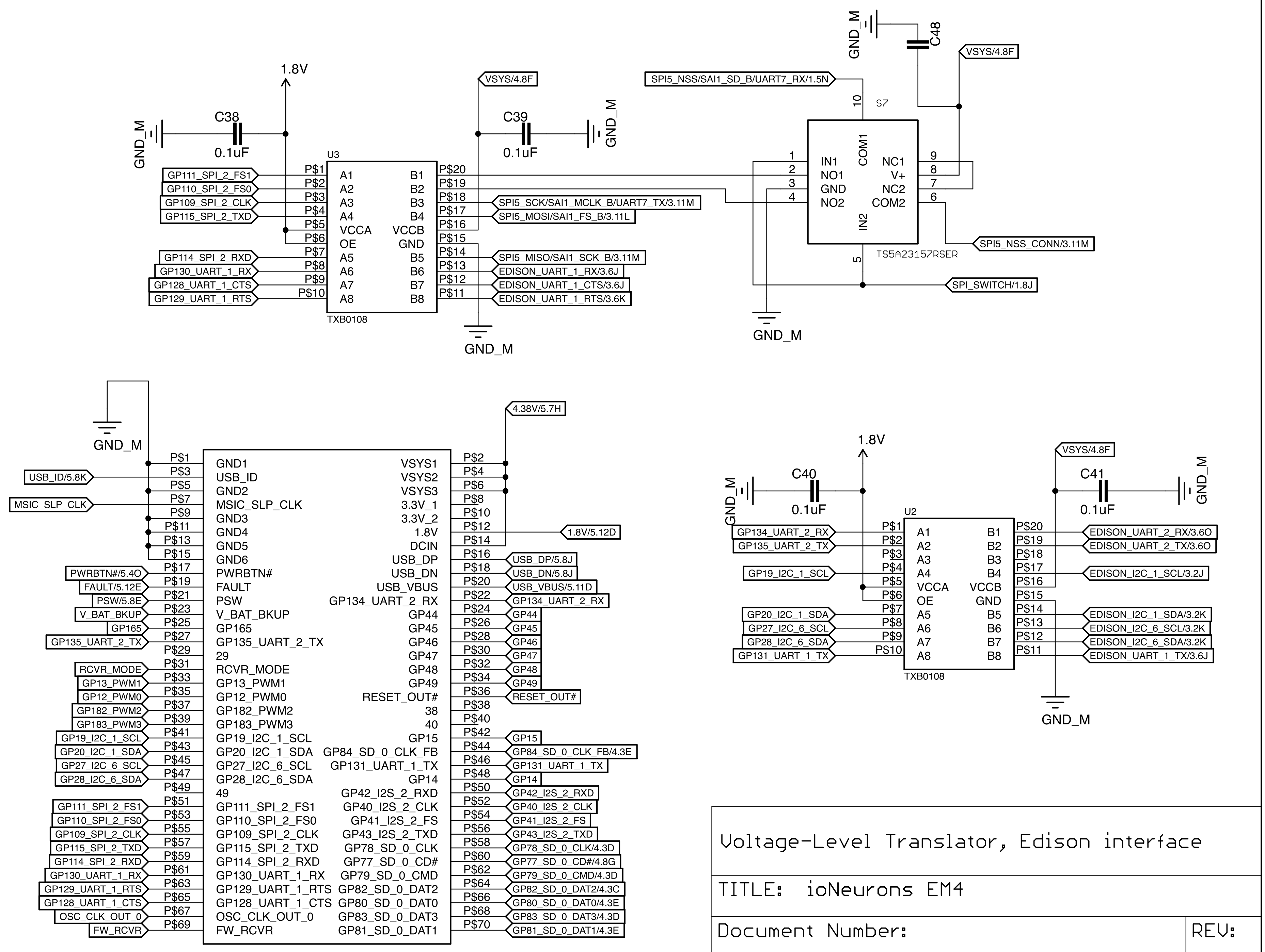
TITLE: ioNeurons EM4

Document Number:

REV:

Date: 2017/09/20

Sheet: 5/11



Voltage-Level Translator, Edison interface

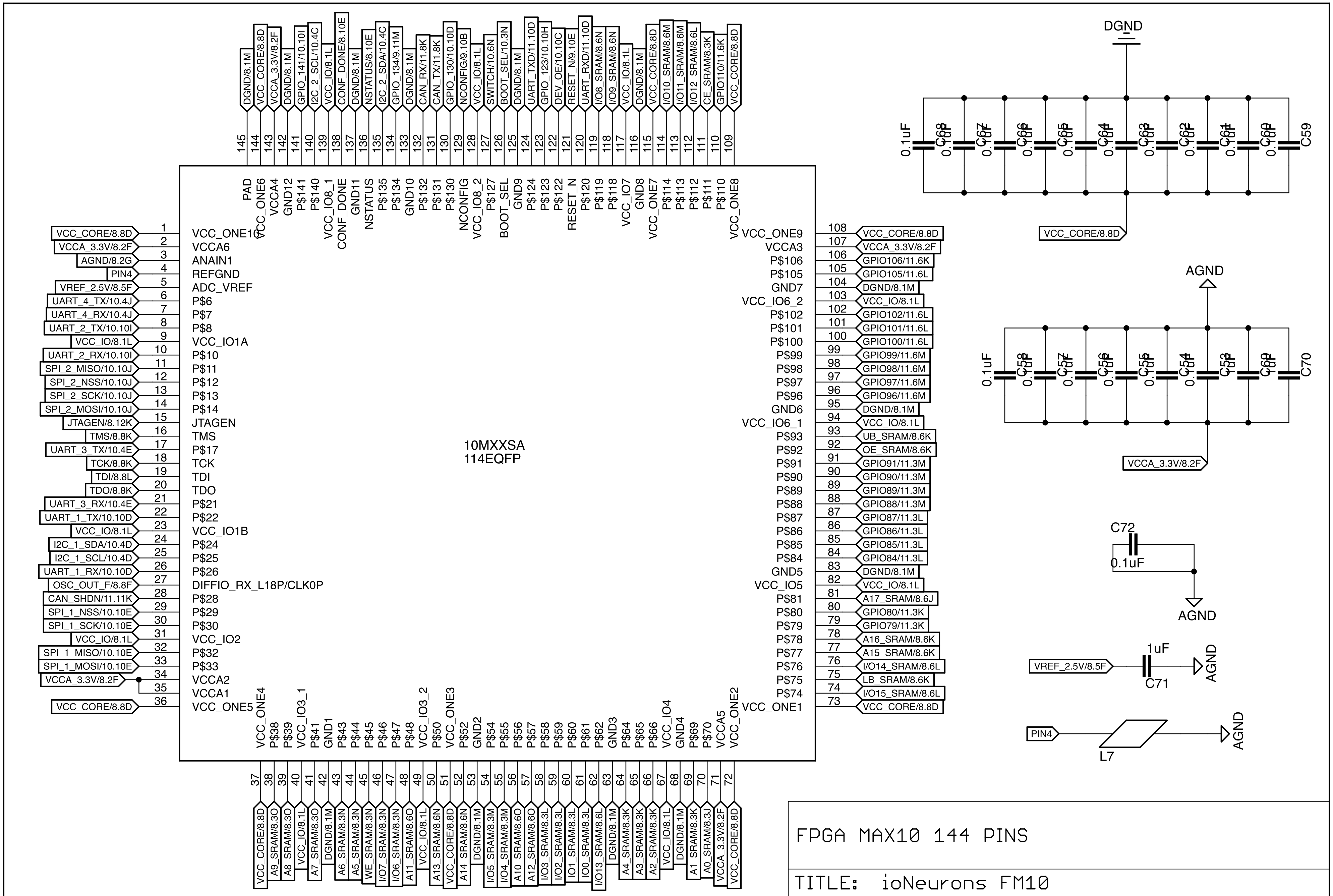
TITLE: ioNeurons EM4

Document Number:

REV:

Date: 2017/09/20

Sheet: 6/11



FPGA MAX10 144 PINS

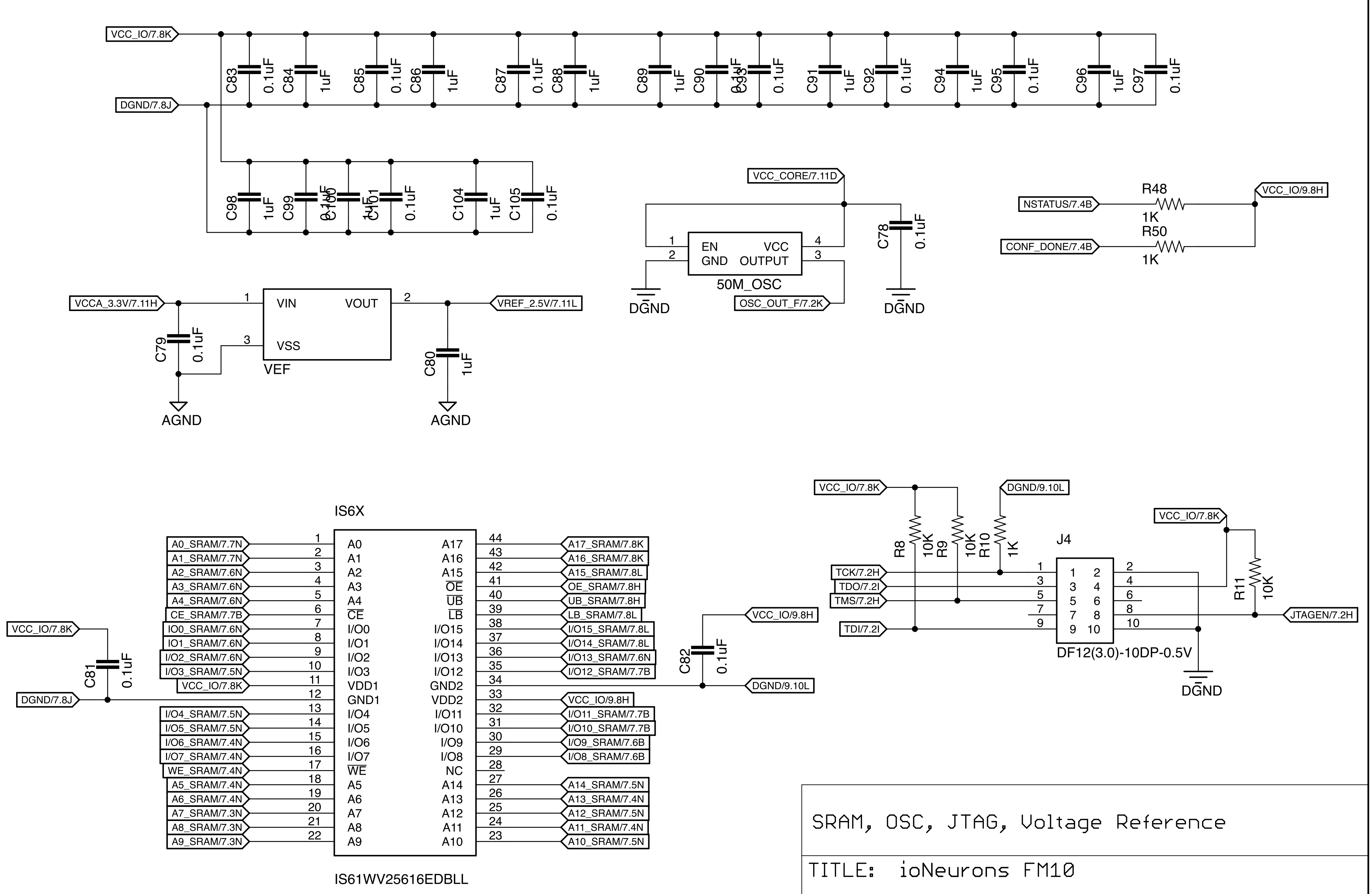
TITLE: ioNeurons FM10

Document Number:

REV:

Date: 2017/09/20

Sheet: 7/11



SRAM, OSC, JTAG, Voltage Reference

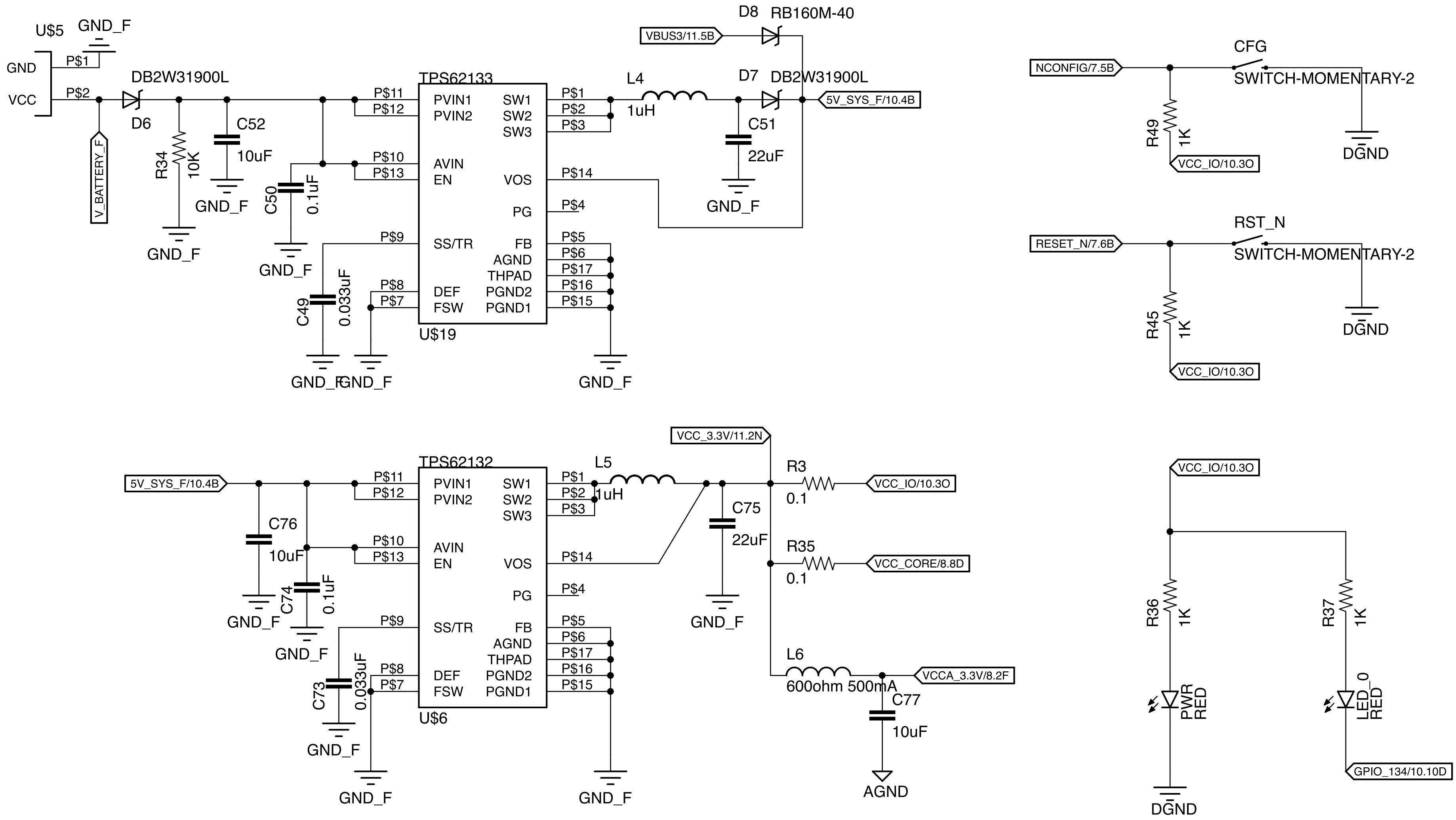
TITLE: ioNeurons FM10

Document Number:

REV:

Date: 2017/09/20

Sheet: 8/11



Power (IO, Core, Analog)

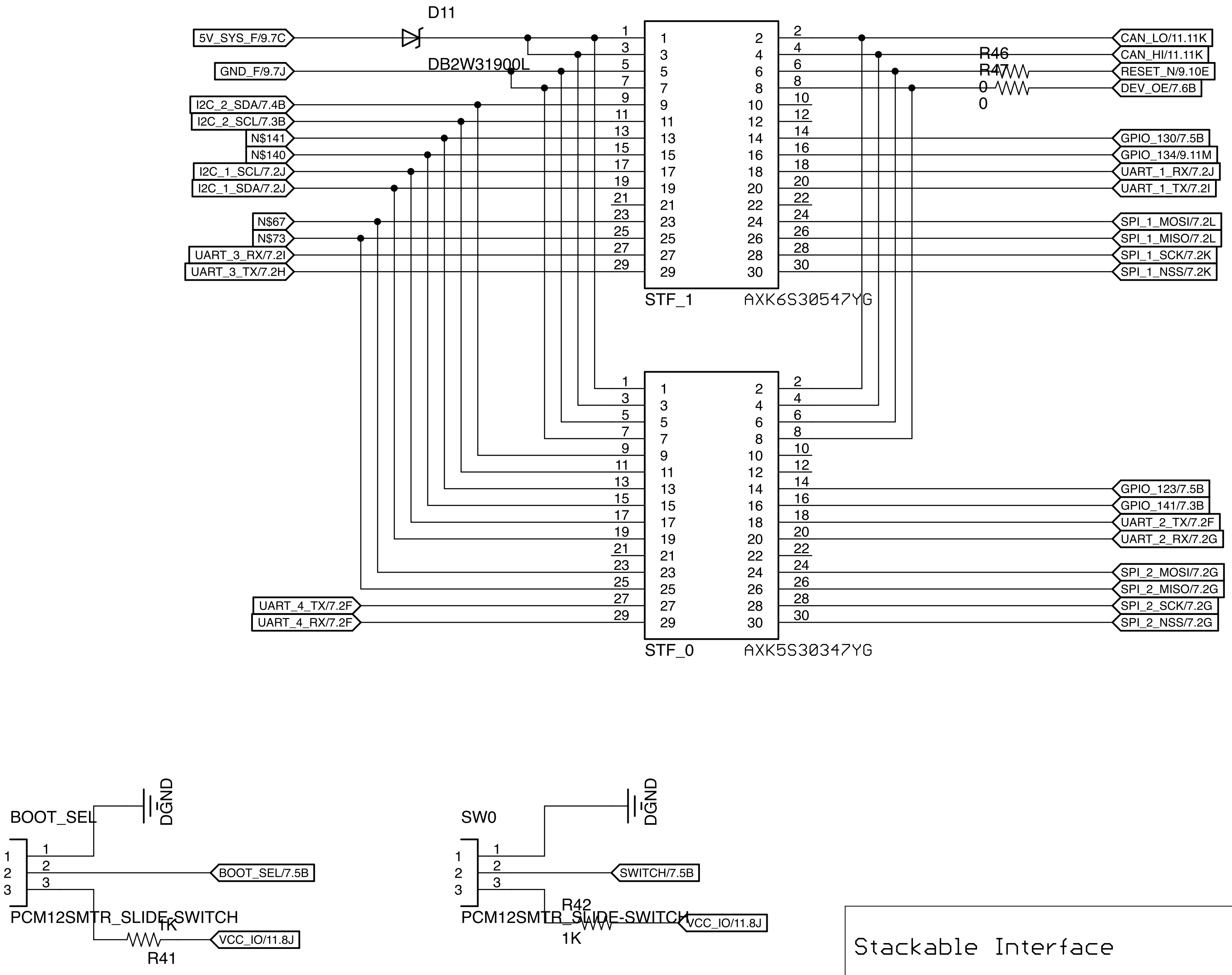
TITLE: ioNeurons FM10

Document Number:

REV:

Date: 2017/09/20

Sheet: 9/11



Stackable Interface

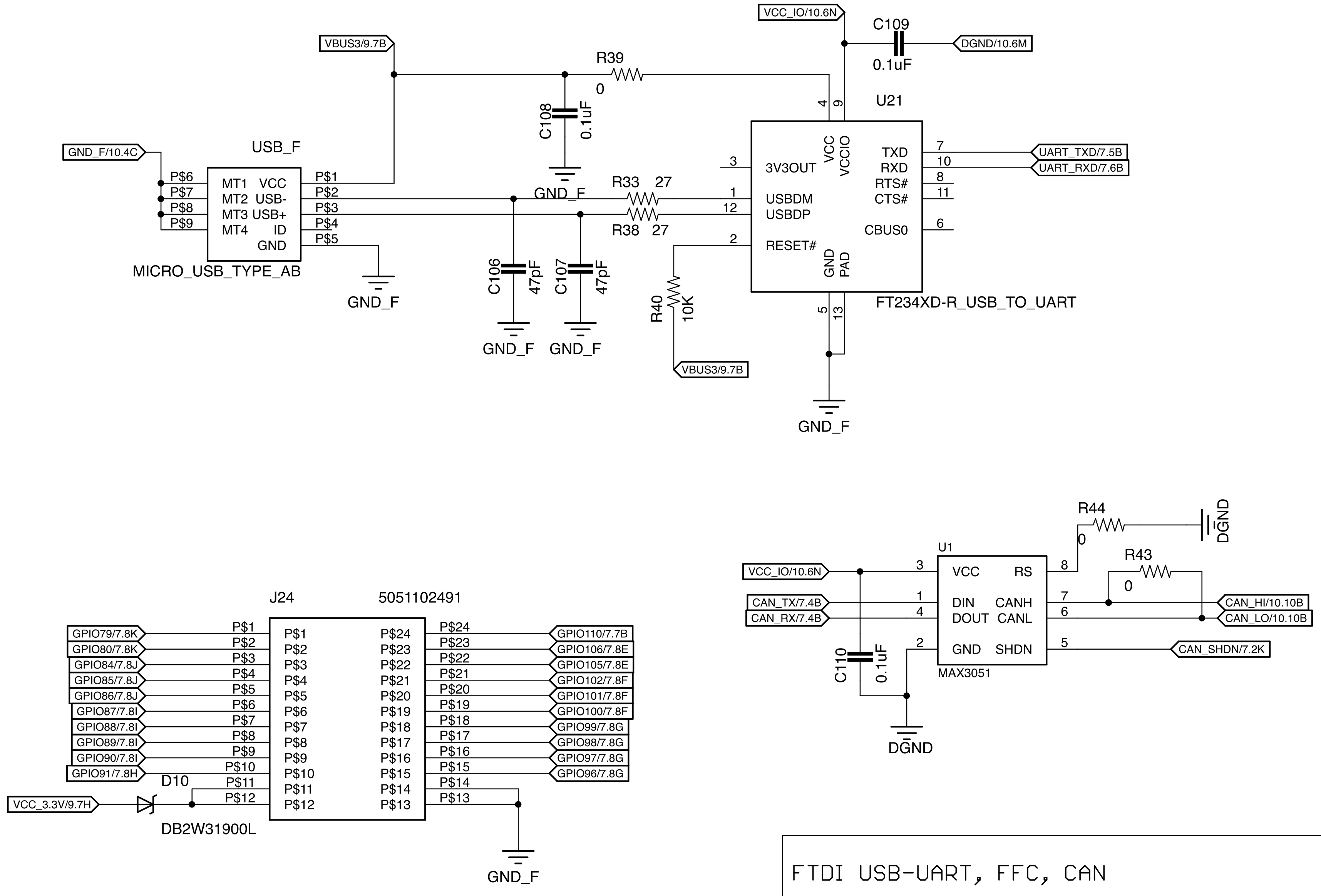
TITLE: ioNeurons FM10

Document Number:

REV:

Date: 2017/09/20

Sheet: 10/11



FTDI USB-UART, FFC, CAN	
TITLE: ioNeurons FM10	
Document Number:	REV:
Date: 2017/09/20	Sheet: 11/11

Bibliography

[1] Z. Farooq, U.; Marrarkchi and H. Mehrez. Tree-based heterogeneous fpga architectures. *Application Specific Exploration and Optimization*, XVI:7 – 23, 2012.

[2] Bettina Weiss Gunther Gridling. *Introduction to Microcontrollers.* 2007.

[3] Ron Wilson. In the beginning. *Altera.com*, 2013.

[4] Wim Roelandts. 15 years of innovation. *X Cell, The Quarterly Journal For Programmable Logic Users*, 32, 1999.

[5] Motorola Inc.; G.Viot R. Bannatyne. Introduction to microcontrollers. i. *Wescon/98. Conference Proceedings*, 1998.

[6] Steve Heath. *Embedded Systems Design.* Newnes, 2003.

[7] Mohammed Raju Hossain; Taufiqur Rahman; and Nicholas Krouglicof. An agile single board quadrotor providing “eye in the sky” capabilities for marine environments. In *Oceans - San Diego.* IEEE, 2013.

[8] Christophe Meynard Ahmad Audi, Marc Pierrot-Deseilligny and Christian Thom. Implementation of an imu aided image stacking algorithm in a digital camera for unmanned aerial vehicles. *Sensors (Basel)*, 17(7):1646, 2017.

[9] Nicholas Krouglicof. Rigid-body pose measurement from a single perspective view. In *Intelligent Autonomous Systems, IAS–3: An International Conference, Pittsburgh, Pennsylvania, February 15-18, 1993*, page 368. IOS Press, 1993.

[10] A. Rosenfeld and A. C. Kak. Digital picture processing. *Academic Press, Incorporated*, 1, 1982.

[11] Zoran Salcic. Protos - a microcontroller/fpga-based prototyping system for embedded applications. *Microprocessor and Microsystems*, 21:249 – 256, 1997.

[12] Zoran Salcic. Fpgas and cplds - a challenge for complex digital system design. *IPENZ'97 Conference, Wellington*, 1997.

[13] S.H.M. Ludwig. The design of a coprocessor board using xilinx's xc6200 fpga. In *FPL '96, Darmstadt, also Lecture Notes in Computer Science*. Springer, 1996.

[14] Intel Corporation. Intel max 10 power management user guide. UG-M10PWR, 2017.

[15] OmniVision Technologies. Ov5640 color coms qsxga image sensor with omnibsi technology. *Datasheet(CSP3) Product Specification*, page 19, 2011.

[16] InvenSense Inc. Mpu-9250 product specification. *Datasheet*, (PS-MPU-9250A-01):5, 2016.

[17] STMicroelectronics. Vl53l0x world smallest time-of-flight ranging and gesture detection sensor. *Datasheet - Production Data*, page 1, 2016.

[18] Rajan Arora. I2c bus pullup resistor calculation. *Application Report SLVA689*, 2015.