Ho Chi Minh City University of Technology

Smart Home

Student: Tuan-Hung VU
ID: 1450231

Day Month Year

Abstract

Abstract goes here

Acknowledgements

I want to thank...

Contents

List of Figures

List of Tables

Acronym and Abbreviation

ADC	Analog to Digital Converter
ALU	Arithmetic Logic Unit
CPU	Central Processing Unit
CU	Control Unit
DAC	Digital to Analog Converter
MCU	Microcontroller Unit
MQTT	Message Queue Telemetry Transport
RAM	Random Access Memory
ROM	Read-only Memory

Chapter 1

Introduction

asdasdasdasdasdas

Chapter 2

Background

2.1 Microcontroller

2.1.1 Theory

Microcontroller Unit (MCU) is a small size, special purpose computer. It is small enough in order to be integrated on a small circuit in which will do specified tasks or applications. MCU itself comes with memory, input, output peripherals and processor. Program to run the MCU is stored in Read-only Memory (ROM) and usually not change in production. A microcontroller is usually designed to run in small size and at low cost, which is compatible to be embedded in other system in order to control actions of the system automatically.

Few advantages of MCU over a microprocessor can be listed as following:

- A MCU is already a standalone microcomputer.
- Because it can be considered as an independent computer, most needed components are integrated on a small size board.
- The above reason leads to the benefit that using MCU can make the system compact, highly mobile and cost efficiency.
- Time reduction because it is programed to run specified set of commands only.
- It is also easy to use and maintain.
- MCU nowadays usually designed to be used with low power in order to last longer under energy-limited condition.

2.1.2 Microcontroller structure

Figure 2.1 demonstrates the basic structure of a microcontroller. It is easily to see the basic design of a microcontroller and its components.

- CPU: is the central unit which is assembled with Arithmetic Logic Unit (ALU) and a Control Unit (CU). Its functions are connect parts of the MCU into a single system by doing fetch, decode and execution.

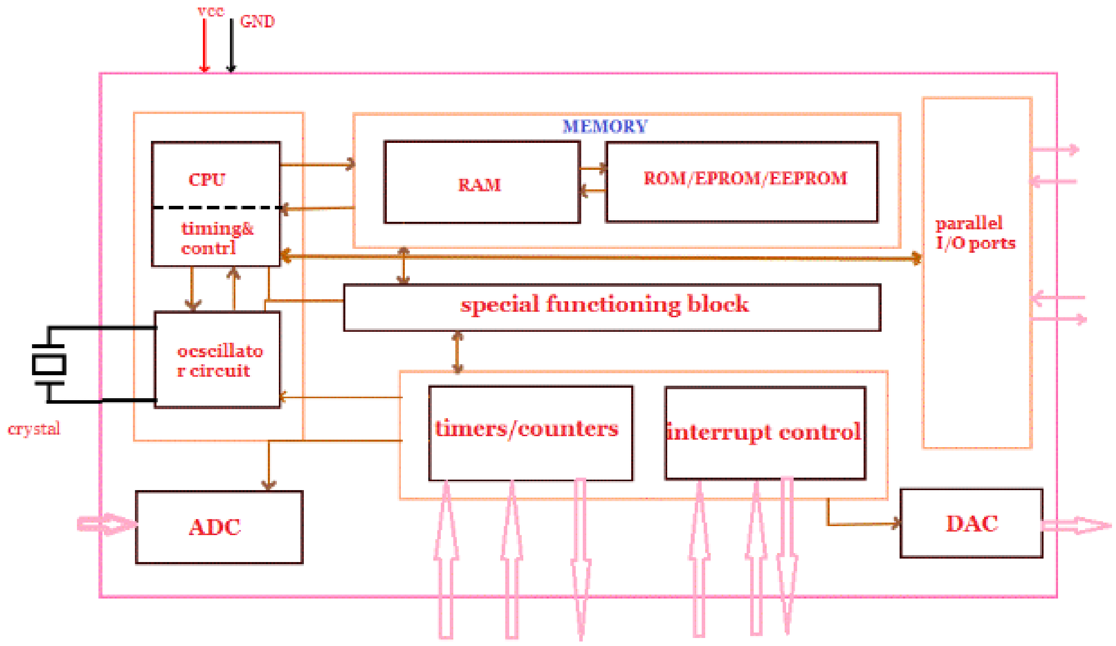

Figure 2.1: Structure of Microcontroller

- Memory: there are two types of Memory that are required, namely ROM and Random Access Memory (RAM). Each type has its own functions, in which ROM will handle the program and the written instructions and RAM can only store temporary data while the program is executing.
- Input/Output: the single board system needs input to execute the program as well as outputs to delivery the information for further handling. The I/O peripherals are the interface of the MCU to communicate with or to control other devices.
- Bus: bus is the system of wires that used to connect the Central Processing Unit (CPU) with other peripherals, which means it plays an important role but rarely discussed.
- Timers/Counters: they are built-in components for microcontroller, which is used to count in order to handle external events.
- Interrupts: is used to interrupt that can be an external or internal one, which helps to execute an instruction(s) while the main program is executing.
- ADC: Analog to Digital Converter (ADC), its name says it all, which is a circuit use to convert analogs signal to digital signals. The reason to use ADC is most sensors available on the market can read only analog signal but CPU of the MCU can read digital signal only, so a ADC is necessary for them to communicate.
- DAC: Digital to Analog Converter (DAC) similar to ADC, DAC is also a circuit which convert digital signals into analog signals for further processing.

2.1.3 Microcontroller market

There exists many microcontrollers on the market which come in various sizes and capacities. The list is only containing very few popular MCU that the author knows of.

- Intel 8051
- STMicroelectronics STM8S (8-bit), ST10 (16-bit) and STM32 (32-bit)
- Atmel AVR (8-bit), AVR32 (32-bit), and AT91SAM (32-bit)
- Freescale ColdFire (32-bit) and S08 (8-bit)
- PIC (8-bit PIC16, PIC18, 16-bit dsPIC33 / PIC24)
- Renesas Electronics: RL78 16-bit MCU; RX 32-bit MCU; SuperH; V850 32-bit MCU; H8; R8C 16-bit MCU
- PSoC (Programmable System-on-Chip)
- Texas Instruments Microcontrollers MSP430 (16-bit), C2000 (32-bit), and Stellaris (32-bit)

2.2 Communication

2.2.1 Introduction

Nowadays, there are various communication protocols can be used for the thesis, namely I2C, ISP, RS232, RS-485, Bluetooth or Wi-Fi. Each protocol is designed to be suitable for specified purpose with different advantages or disadvantages, which means a perfect protocol does not exist. When making a decision to choose suitable protocols for the thesis, the trade-off between the stabilization and the speed of the communication protocol has been considered carefully.

In this thesis, RS-485 is chosen as the main way for components in the system to communicate with each other. RS-485 is defined in 1983 not as a protocol but an electrical interface standard and only specifies the drivers and receivers' characteristics. It is developed in order to make data rate and transmitting distance are inversely proportional. For instance, the data transmitting speed can reach 10 Mbps within distance of 16 meters or if the distance is extended to 1220 meters, the data rate is lower to 100 kbps. The advantage of RS-485 over RS232, which is developed in 1960, is multiple nodes can be parallel connected to a bus. Additionally, the network can be extended in length and number of nodes easily by using simple connectors. Besides, Wi-Fi, Bluetooth, GSM and MQTT are also implemented in the thesis in order to take the advantages of different communication protocols in different circumstances.

2.2.2 Universal asynchronous receiver-transmitter (UART)

2.2.3 RS-485

Table 2.1 shows the remarkable specifications of RS-485. With these characteristics, RS-485 was a robust interface standard and was able to meet the requirements in industries, in which implemented applications that need a stable, fast and reliable connection.

Figure 2.2 demonstrates two ways to implement the connection with RS-485, which are full-duplex and half-duplex. Full-duplex implementations require four-wire (two signal pairs) instead of two-wire in half-duplex implementations; But despite the downside of two-wire implementation is it is limited to half-duplex and needs attention to turn-around delay, in practical applications, half-duplex is most chosen. The reason is full-duplex solution depends on master-slave model, which means the slaves cannot communicate with each other. In modern designs of transceiver, the allowed number of nodes can connect to the bus is up to hundreds.

Name	Detail
Differential	Yes
Number of supported devices	32 transmitters/32 receivers
Operation mode	Half-duplex
Longest supported distance	at 100kbps: 1200 meters
Highest supported transmitting speed	at 10Mbps: 16 meters
Mark (data = 1) condition	1.5V to 5V (A negative towards B)
Space (data = 0) condition	1.5V to 5V (A positive towards B)
Output current capacity	250mA
Receiver input sensitivity	±200 mV
Receiver input range	-7V to 12V

Table 2.1: RS-485 Remarkable Specifications

Working principle of RS-485 is different in comparison to other standard; Instead of using a zero ground as the voltage reference, which will cause noise over the communication length, it uses floating voltage between two wires of the signal pairs, A and B or (+) and (-). After transmitting, the receiver compares the different of voltage between two wires and achieved the correct data with the lowest noise may cause. Figure 2.3 illustrates an example of the RS-485 waveforms transmitting one byte which has Mark, Space, and Idle phases. In most network, there will be one node acting as the master and the rest work as the slaves. At this point, the master sends command frame over the connection, and all slaves receive the data, then each slave with different functionality will work as programmed with different received data and also response to the master as programmed. The best practical result is obtained with the use of twisted pair of wires because some of the noise current will flow in the opposite direction with the current in the cable. In case using the straight cable, the noise current flows straightly along the cable in the same direction which will cause a loop current. Combined with the twisted pairs of wires, the cable also comes with shield, which is an accepted approach to restrain the noise, is used in applications that need higher noise resistance.

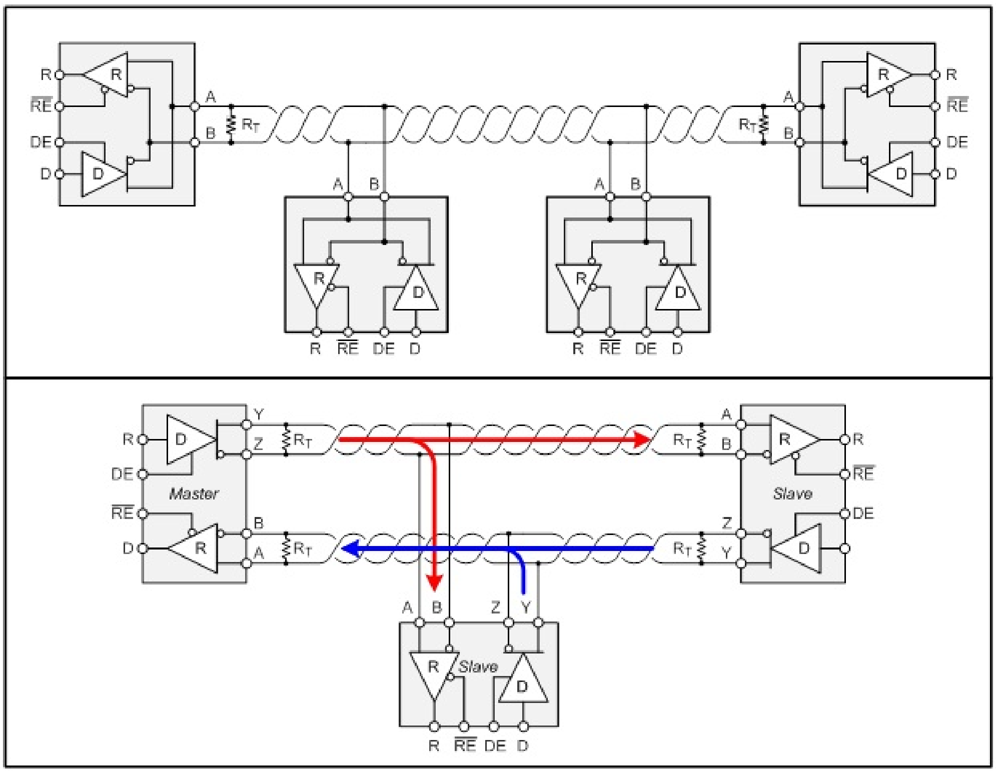

Figure 2.2: Half-duplex and Full-duplex implementations

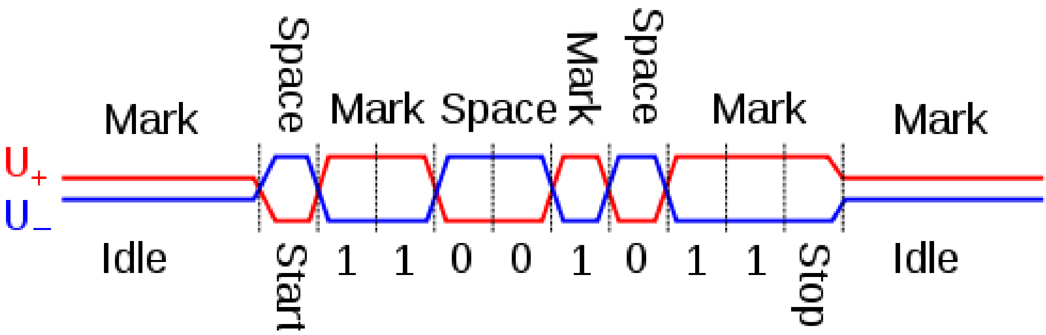

Figure 2.3: RS-485 waveform

As written in the introduction, RS-485 can connect with multiple transmitters and receivers in the same network. For instance, using input resistance around 12kΩ,the numbers of devices can connect to the system is up to 32. Besides, with the connecters, this number can increase significantly to the number of thousands and the transmitting distance can be also extended to kilometers. In addition, the network implemented with RS-485 needs termination, usually with a 120Ω resistance at the end of two wires. This is applied to terminate or minimize the reflection in order to avoid the fraud of sending data. Furthermore, it usually included pull up and pull down resistors for fail-safe bias in each wire in case that any wire is not controlled by any device. When input voltage ranging from -200mV to +200mV, receiver understands as "undefined" state, which caused by several reasons such as system is shutdown, connection from receiver to network is lost, or cable has an open or short part. In this case, fail-safe biasing is applied in order to confirm that the receivers receive defined states only.

2.2.4 MQTT

MQTT is abbreviation of Message Queuing Telemetry Transport, which is a protocol laid in Application layer of OSI model. It is designed as a machine-to-machine and remarkably lightweight protocol that helps communication between constrained devices becomes effortless in comparison to other wireless protocols. In detail, its working principle based on publish and subscribe methodology in order to reduce the amount of transmitting data which leads to the reduction of used bandwidth, latency and power consumption.

A Message Queue Telemetry Transport (MQTT) system is the combination of a server, but usually named broker, and the clients, in which can acts as either a publisher, subscriber or both. One broker can have numerous clients connect to and each client can subscribe to any topic it is programmed. These subscribers are following and watching for the changes of data of the subscribed topics, once other clients which are defined as the publisher publishes message to the topics, then the broker distributed the payload of message to other clients who had subscribed to those topics. In this scenario, the publisher and subscribers do not need to know the information of each other, the only needed are the topics for publishing and subscribing. Figure 2.4 refers a simple example of a MQTT system consists of one broker and four clients, in which three clients are the publishers and one is the subscriber that subscribes to three topics the publishers publish to. To be specific, three publishers are the sensor nodes publish to three topics, temp1, temp2 and temp3; the last client named Sensor Data Gatherer acts as the subscriber that follows three topics mentioned previously. At once, the subscriber will receive the message whenever a publisher broadcasts the message to any of those topics. Implemented MQTT system will be described in details in later part of the thesis.

2.2.5 WebSocket

WebSocket was introduced in 2011 in term of a communication protocol, implementing full-duplex channels over one TCP connection. The protocol is laid in Application layer of OSI model but distinguished from HTTP. Distinct to HTTP, in WebSocket applications, a server can send the data to client without waiting for requests from client. All data are being sent between server and client will be sent to a settled connection which helps accelerate the data rate and keep the connection opened when necessary. Similar to MQTT, it is designed to reduce the transmitting data, which leads to the reduction of bandwidth and consumed power. Although it is designed for web applications, still, it can be implemented in any applications that need such a lightweight protocol. Figure 2.5 illustrates how WebSocket works. There are two parts of the protocol, handshaking and transmitting data. At first, client sends a request to server to initialize the websocket connection, the server then send an acceptance to connect. After this point, the data are being sent as WS frame with numbering as in the Figure 2.5.

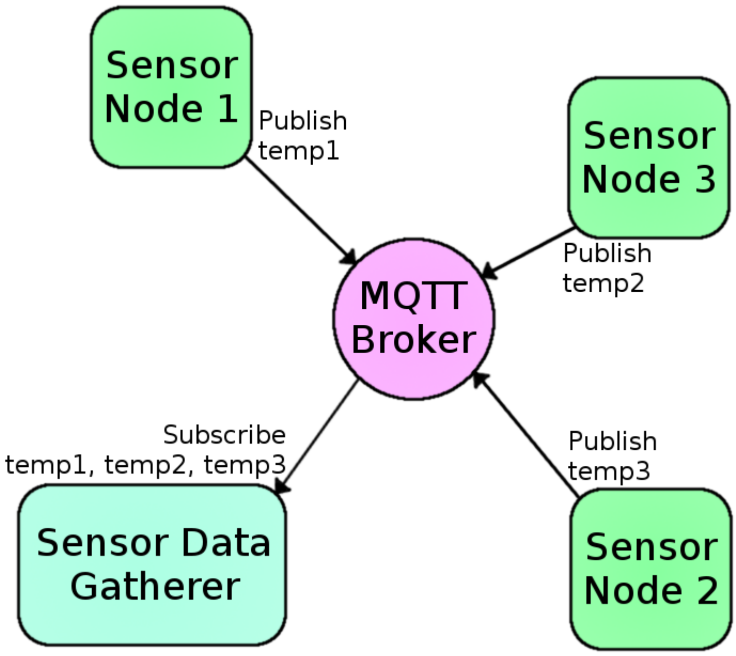

Figure 2.4: Simple example of a MQTT system

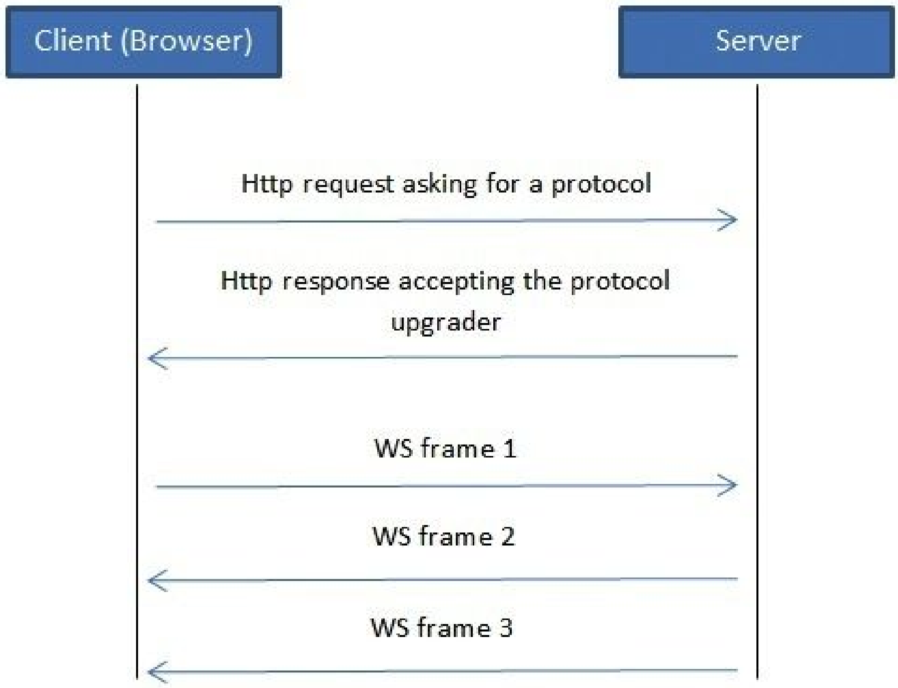

Figure 2.5: WebSocket working principle

2.3 Facial Recognition

2.4 Application Server

Chapter 3

Hardware Design of The System

3.1 Expected System Diagram

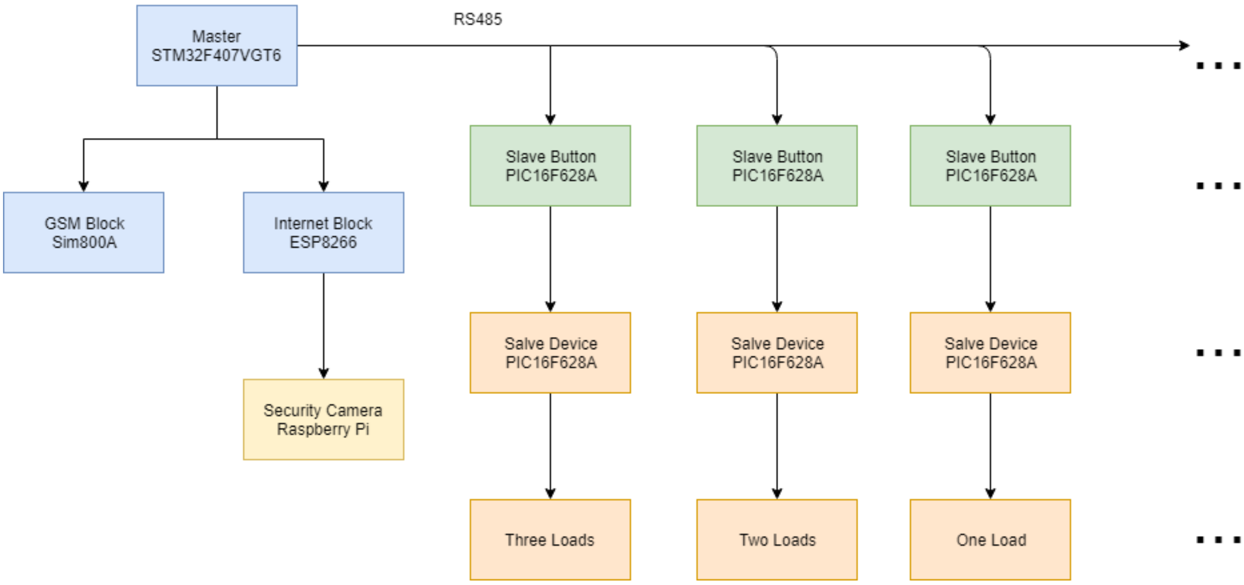

Figure 3.1: Expected hardware blocks

Attributes	Detail	Notes
Maximum number of devices	99	Can be extended by extending command frame
Longest distance supported	1200 meters	
Communication method	Mainly RS-485	
Wi-Fi connection	Wi-fi 2.4GHz	

Table 3.1: System ideal characteristics

In Figure 3.1, there are blocks named Master, Slave Buttons, Slave Devices, GSM block, Internet block and Security Camera. Each block is indicated with implemented hardware and how they connect to each other. As in the figure 3.1, Master is connecting with number of Slaves by UART over RS-485; Also, Master is implemented with GSM and Internet blocks in order to help end user controlling devices and receiving alerts over GSM or Wi-Fi. Each slave connects to the system has the same working principle but different names. In this thesis, there are two slave-2-devices and one slave-3-devices alongside with two slave-2-buttons and one slave-3-buttons to control the loads, respectively. Besides, the author designed one slave-2-buttons and one slave-1-button to control three out of seven existed devices. Three slave-devices are implemented with relays switch state for devices in the house, last device (Device 3) of slave-3-devices is assigned as the Main Door trigger to demonstrate the Security Camera System with Facial Recognition later on.

Internet Block is the middle man for communicating between Application Server and the System. With this block implemented, end-user can control devices without pushing the physical buttons, which may causes difficulties for users because the owners can control their house whenever and wherever they want. Besides, with the help of the Application Server, end-users can collect and monitor data in the house in order to diagnostic and maintain precisely. GSM block should be installed in order to help in the event that Internet block is having unexpected problems.

Security Camera block is the block that monitors the main door and inside the house. The camera installed outdoor is responsible for outdoor security in which it will track people entering the house with a facial recognition system. Additionally, indoor camera should handle the motion detection system while the owner is not at home in order to find strange motion which maybe a burglar breaking in the house. These two system will track and alert by emails, mobile application and text message over GSM network in the case that they detect something. Furthermore, the three-dots indicates that the system can be extended with number of slaves over RS-485, but only up to 99 dues to the limitation of command frame.

3.2 Master Design

3.2.1 Microcontroller Requirements

There are few requirements for the Microcontroller that the author decided to build the system for the thesis, listed as following.

- Support UART in order to communicate with other modules, namely RS-485, ESP8266 and SIM800A.
- Has widely support community.
- Easy to learn to program.
- Extendable with installed components.
- Price and ability for effortless replacement.

Based on the requirements, the chosen MCU is STM32F407VGT6 with STM32F4 Discovery Kit from STMicroelectronics. Figure 3.2 refers the real kit in the market. It is considered as a suitable MCU because of the following reasons.

- The board has large support community.
- Programmed with C language with countless documents.

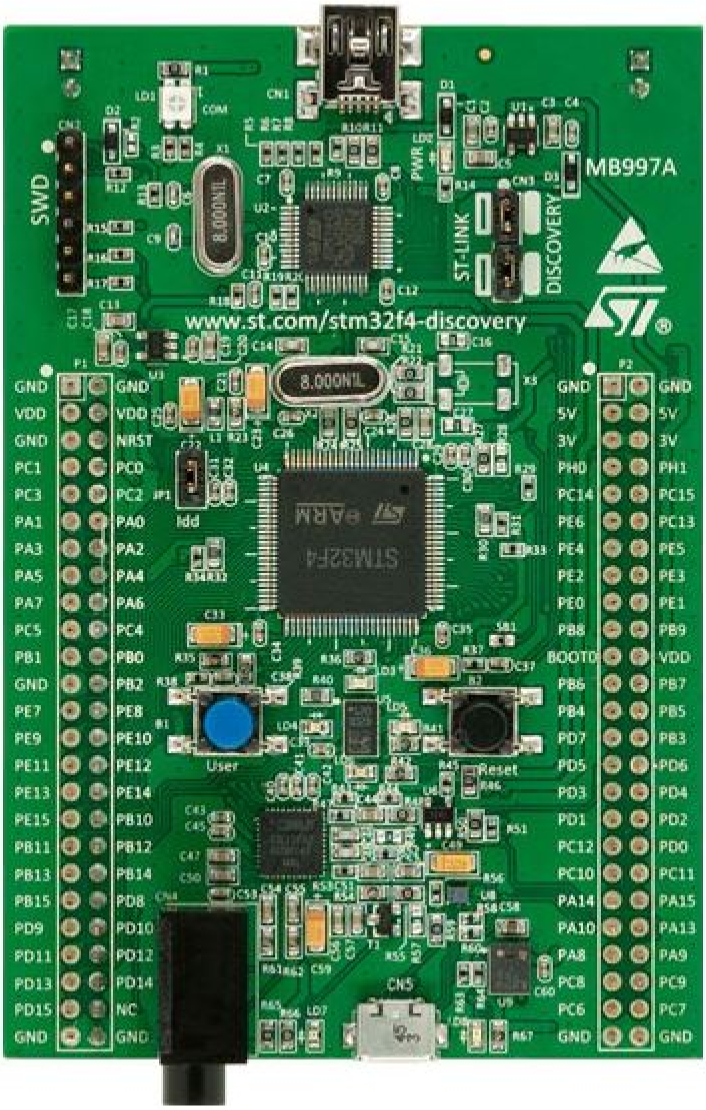

Figure 3.2: STM32F4 Discovery Kit

- MCU used is STM32F407VGT6, with core ARM Cortex 32bit M4, clock up to 168Mhz.
- Support up to 140 I/O.
- Flash memory 1MB.
- Easy to flash even with end-user.
- Cheap price and easy to find replacement parts.

In this thesis, to ensure the effortless replacement of the system parts, the author designed with modules attached on PCB by using headers. With this method, whenever an error occurs to any part of the system, end-user can replace the broken part easily without replacing the whole system. Figure 3.3 shows the headers on Master board for STM32F4 Discovery Kit which is chosen for the thesis. In addition, it shows the connection pin of the MCU with other modules over UART. To be

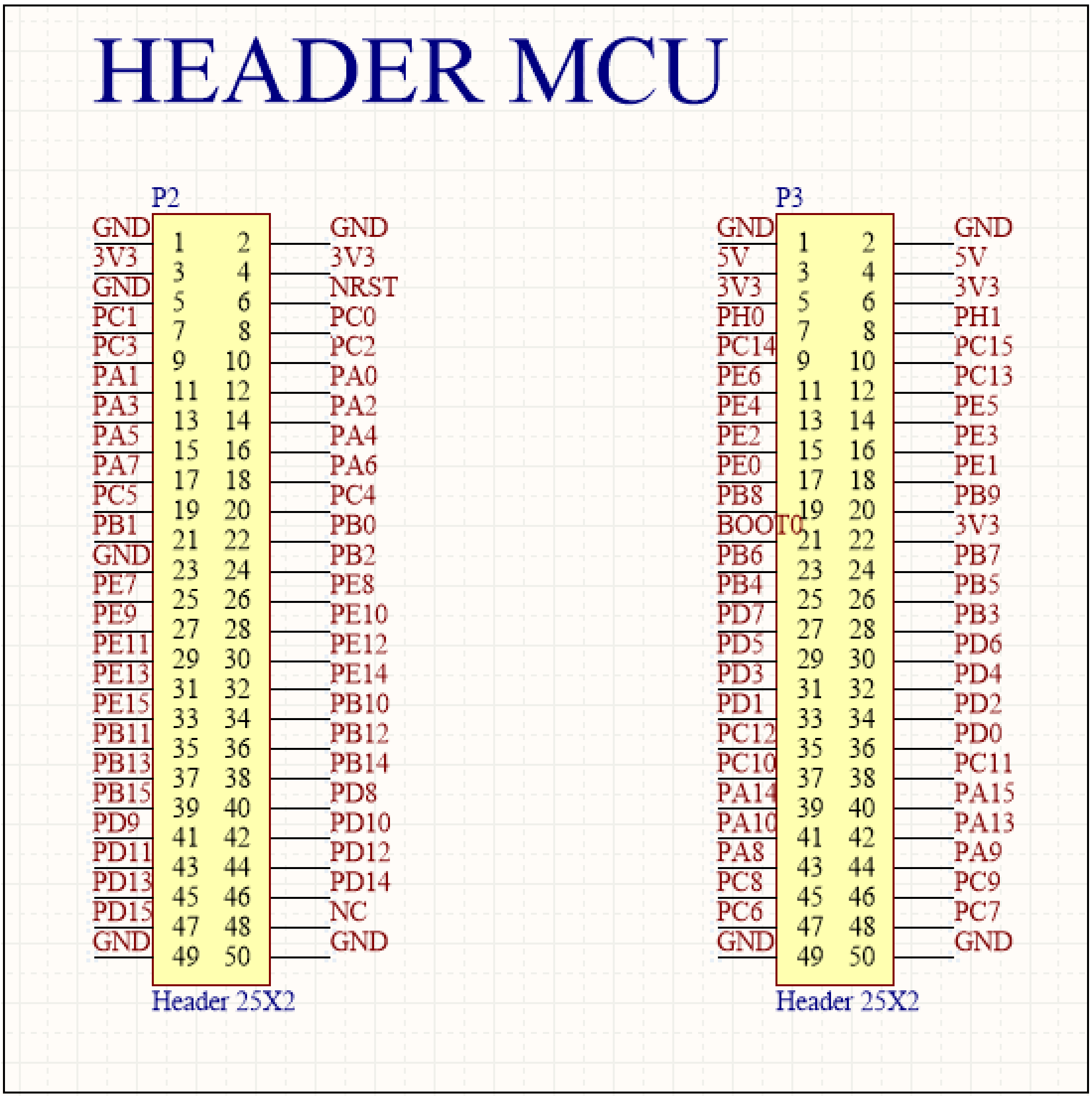

Figure 3.3: Header for STM32F4 Discovery Kit

more specific, MCU connects with RS-485 module over UART1 via pin PB6-PB7, with module ESP8266 over UART2 via pin PD5-PD6, with module SIM800A over UART3 via pin PD8-PD9.

3.2.2 Module RS-485

Figure 3.5 refers the cheap version of module TTL to RS-485 on the market. It integrated IC MAX485 as the main component and other sub-components included termination resistor. This module is stable enough for the system and easy to replace due to its cheap price but does has a weakness which is if it is broken, end-user cannot know unless further tests on the module is processed. The table 3.2 indicates the pin out guideline to connect with the MCU. According to datasheet of IC MAX485, RE and DE must be connected for the MCU to control the module based on logic level, in which the module is transmitting if the pins are pull up to 1, otherwise it is receiving.

Figure 3.4 shows the headers which are used on Master board for RS-485 module

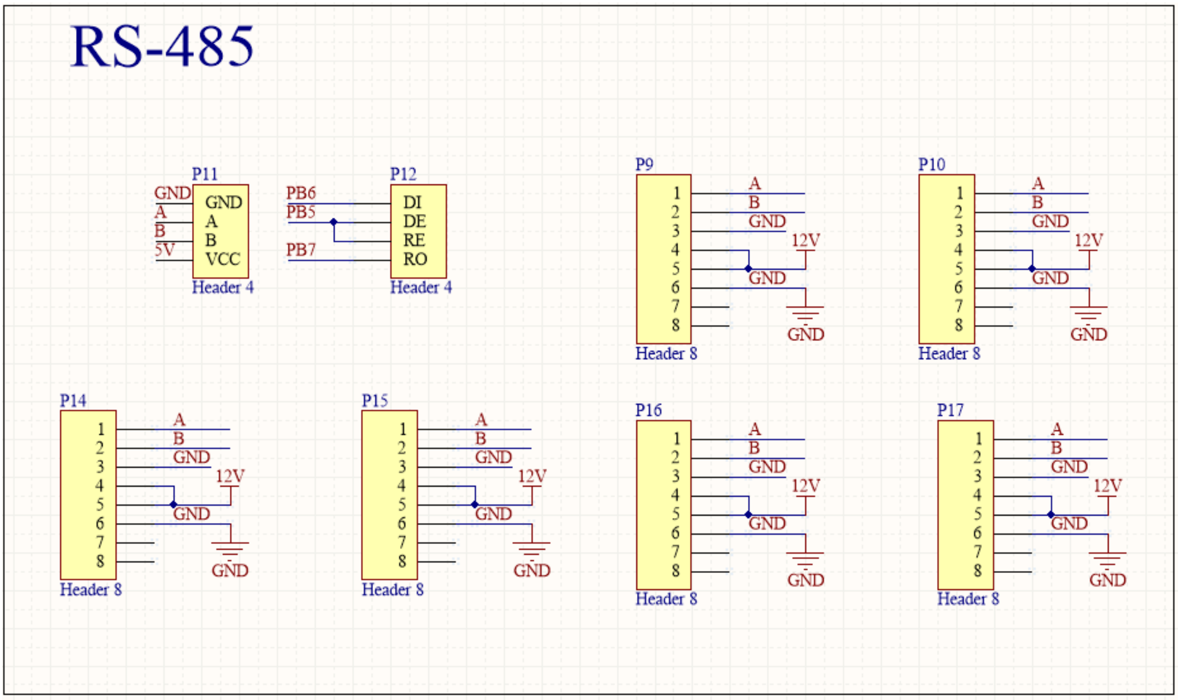

Figure 3.4: Header for module RS-485

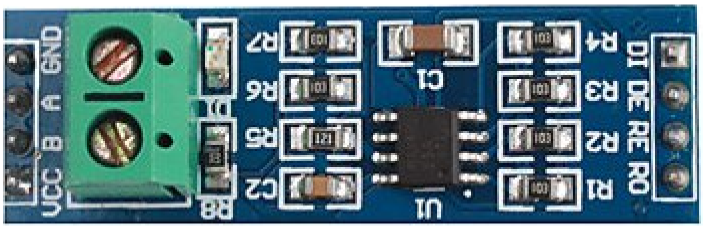

Figure 3.5: Module RS-485

Pin	Detail
VCC	5V
A	Non-inverting Receiver Input and Non-inverting Driver Output
B	.Inverting Receiver Input and Inverting Driver Output
GND	GND, should be 0V
RO	Receiver Output (to Rx pin of microcontroller)
RE	Receiver Output Enable (Low to enable)
DE	Driver Output Enable (high to enable)
DI	Driver Input (to Tx pin of microcontroller)

Table 3.2: Module UART TTL to RS-485 pin out

in figure 3.5 and the headers of RJ-11 female jack for RS-485 output of the Master. The reason for choosing RJ-11 jack and its compatible cable is the cable suits for the project which needs four wires, in which two are the signal wires (A and B of RS-485 standard) and the other two are the pair providing power for other slaves

(12V and GND). With this method, a four-wire twisted cable with shield is used in order to keep the noise as low as possible and still, provides the power along the whole system with only one cable connected.

3.2.3 Module ESP-8266

This module is implemented to establish the connection between the Application Server and the System. End-users can control and monitor their system with a website or an android application over Wi-Fi connection with module ESP-8266. There are various versions of module using ESP-8266 on the market, but the full name of the chosen module is ESP-8266 NodeMCU lua CP2102. It is a small size kit that integrated with ESP8266 SoC, other components and it is also compatible with Arduino IDE which makes it become the easiest to use ESP-8266 module in comparison to other versions.

Attribute	Detail
SoC	ESP8266 Wifi SoC
Firmware	NodeMCU Lua
Flash chip	CP2102
GPIO	compatible with firmware of Node MCU
Power supply	5V DC with Micro USB or Vin
GPIO logic level	3.3V
Integrated LED	Reset, Flash and Status indicator
Dimension	25mm x 50mm
Others	Compatible with Arduino IDE

Table 3.3: Module ESP-8266 NodeMCU lua CP2102 remarkable characteristics

Figure 3.6: Module ESP-8266 NodeMCU lua CP2102

3.2.4 Module SIM800A

3.2.5 Power for Master

In order to provide enough power for every module mentioned above, the author uses a AC/DC adapter with output 12V-5A as the main power supply with a honeycomb power source 12V-3A as a backup one as illustrated in Figure 3.7. In the Power for Master circuit, module LM2596 which is a buck converter, used to convert 12VDC to 5VDC.

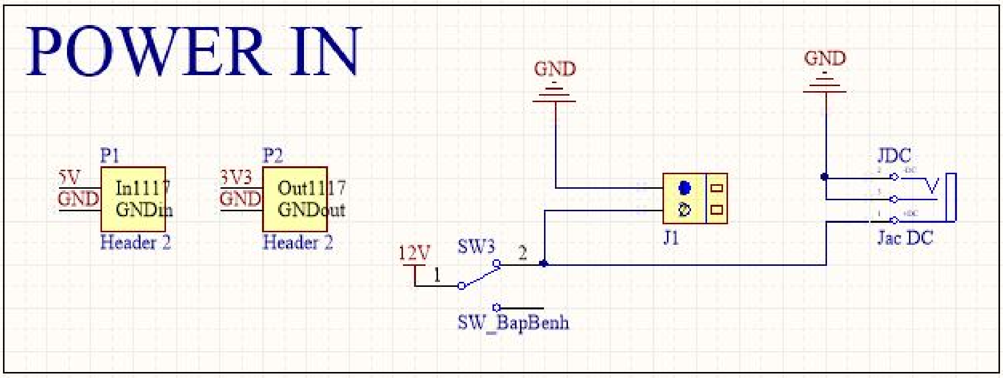

Figure 3.7: Header for Power blocks for Master

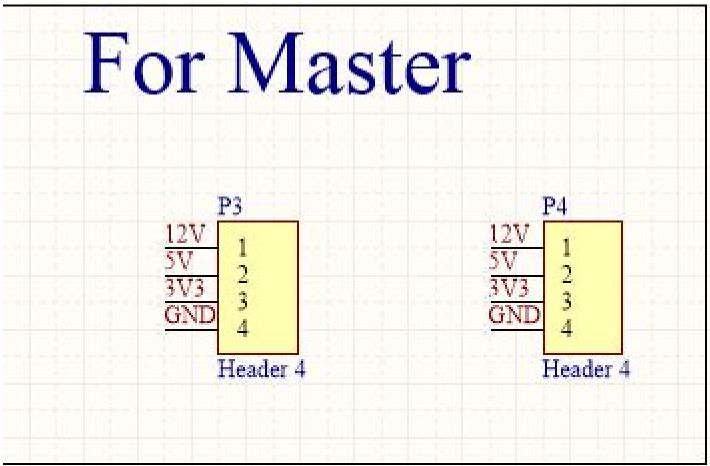

Figure 3.8: Output header of Power for Master

Table 3.4 lists remarkable specifications of module LM2596 using in the project. With these specifications and its cheap price, the module is suitable for various applications, namely voltage dividing, buck converting, supplying for motor, camera or robot. In Figure 3.7, P1 and P2 headers are implemented for module AMS1117. AMS1117 is also a buck converter but from 5VDC to 3.3VDC only. The advantage of this module is that it is integrated in a small circuit (as in Figure 3.10) can supply and maintain maximum current from 800mA to 1A, which is needed for modules that need high current such as module ESP-8266 using in this thesis. Furthermore, the 3.3VDC may be used as a backup power supply for Microcontroller which needs

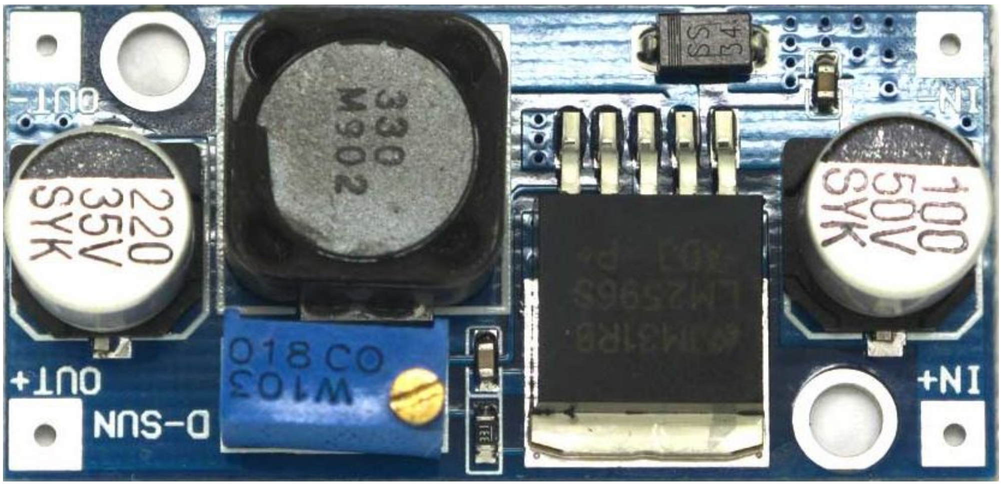

Figure 3.9: Module LM2596

Figure 3.10: Module ASM1117

Attribute	Detail
Input	Ranging 3V-30V
Output	Ranging 1.5V-30V
Maximum current output	3A
Efficiency	92%
Power	15W
Dimension	45mm x 20mm x 14mm

Table 3.4: Module LM2596 specifications

3.3V, which could be extended in further development of the project. Figure 3.8 indicates the output headers for circuit Power for Master, which will supply the Master with three level of voltage source, namely 12V, 5V and 3.3V.

3.3 Slave Design

3.3.1 Requirements

Slave circuits have requirements listed as following.

- Small integrated circuit.
- Support UART to communicate with Master.
- Well documented.
- Large support community.
- Price is cheap.
- Easy to implement or replace when broken.

Figure 3.11: Microchip PIC16F628A

The author chose PIC16F628A as the microcontroller for Slave Buttons and Slave Devices because of its small size, easy to implement or replace and reasonable price. Please see Table 3.5 for the highlight specifications of MCU Microchip PIC16F628A.

Attribute	Detail
Supply power	Ranging 2V-5.5V
Number of pins	18
RAM	224 bytes
EEPROM	128 bytes

Table 3.5: PIC16F628A Highlight Specifications

3.3.2 Power for Slaves

In comparison with Master, power supply for Slaves requires less criteria. There are two blocks of power will supply for slaves in this thesis. In the first design, the author built power block separately from the circuit, but in second design, the power supply for the Slaves is integrated in the same circuit in each Slaves. Please see Figure 3.12 and Figure 3.13 for two blocks that supply power for Slaves in the first and second design, respectively. In first design, power block use the same buck converter LM2596 and ASM1117 as mentioned in section 3.2.5 Power for Master, but in second design, the author chose IC 7805 for all power blocks using in all later Slaves. First design applied for two slaves, namely Slave-3-Relays and Slave-3-Buttons, the second design implemented on all later Slaves, namely Slave-2-Relays, Slave-2-Buttons and Slave-1-Button.

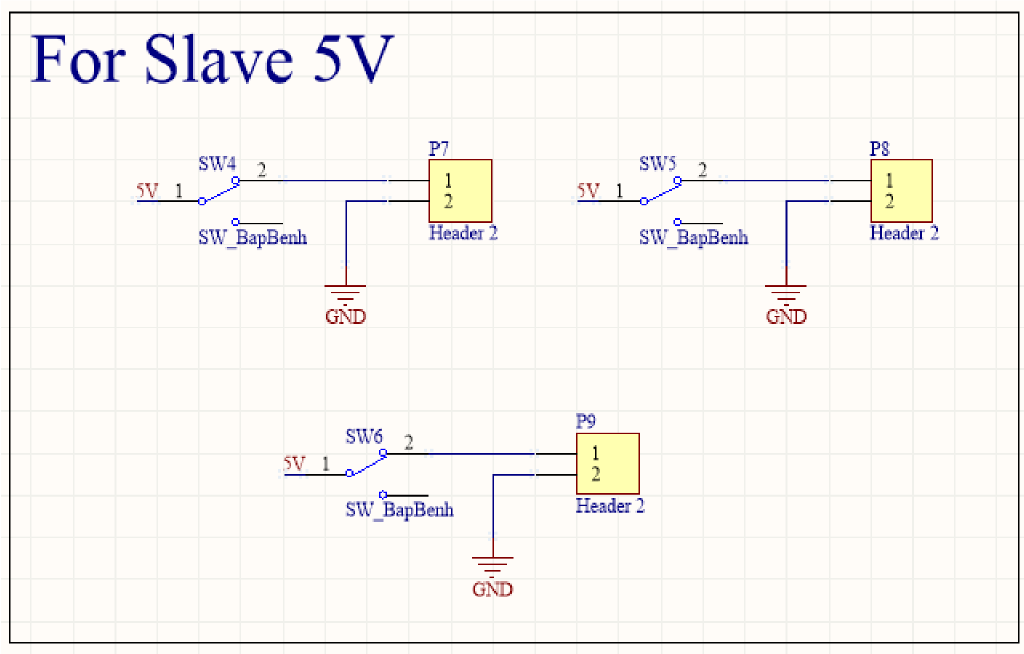

Figure 3.12: Power supply for Slaves 1

3.3.3 Module RS-485

Slaves receiving from and transmitting to Master over RS-485 block. In Slave design, the author uses the same module as in Figure 3.5, the headers for Module RS-485 is also identical to the headers using in Master circuit, but the headers for the output by jack RJ-11 is reduced to two as in the Figure 3.14.

3.3.4 Controller Block

As mentioned in previous section, PIC16F628A is chosen as the MCU for all Slaves in the system. Figure 3.15 and Figure 3.16 refers the MCU block of Slave Button(s) and Slave Relay(s), respectively, in which connect with external crystal with frequency of 20MHz.

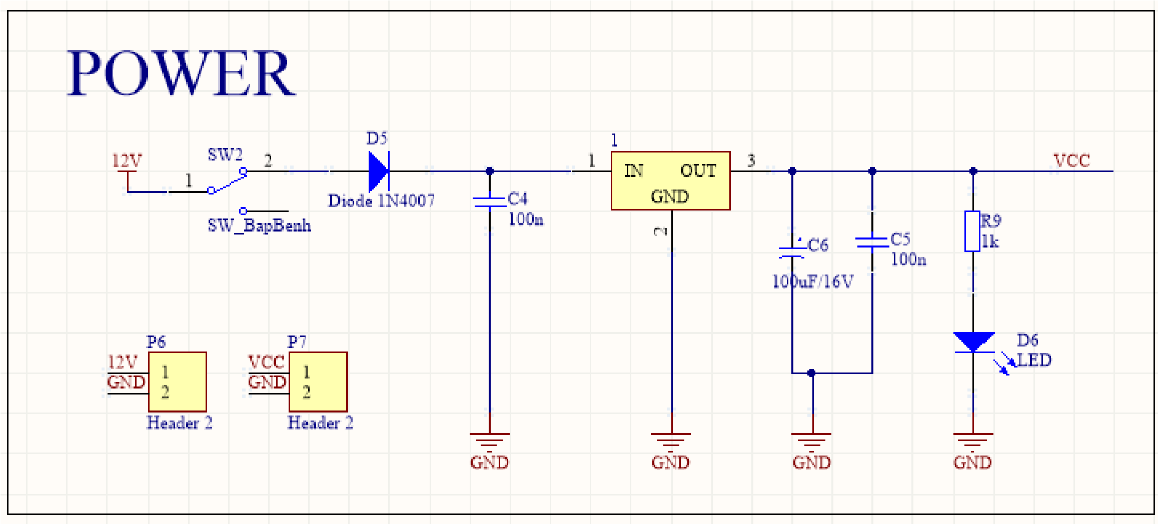

Figure 3.13: Power supply for Slaves 2

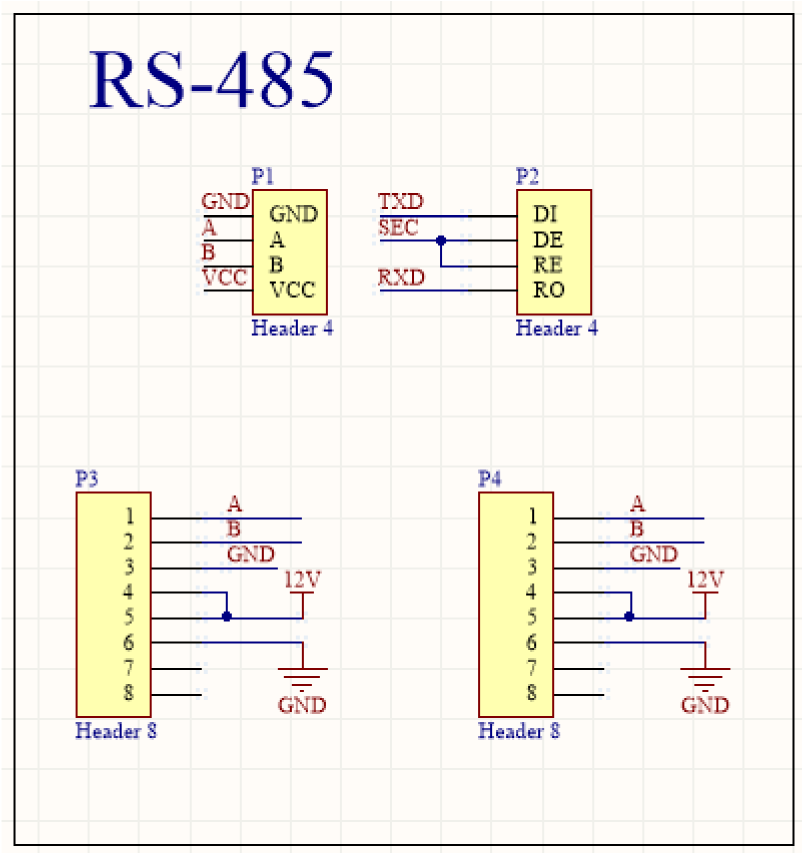

Figure 3.14: Header for module RS-485 in Slave circuits

3.3.5 Button Block of Slave Button(s)

Figure 3.17 sketched the schematic of three-button block of Slave-3-Buttons, which means it is the typical block and may use for different numbers of buttons in one integrated circuit, depends on the decision of the author. In this thesis, the

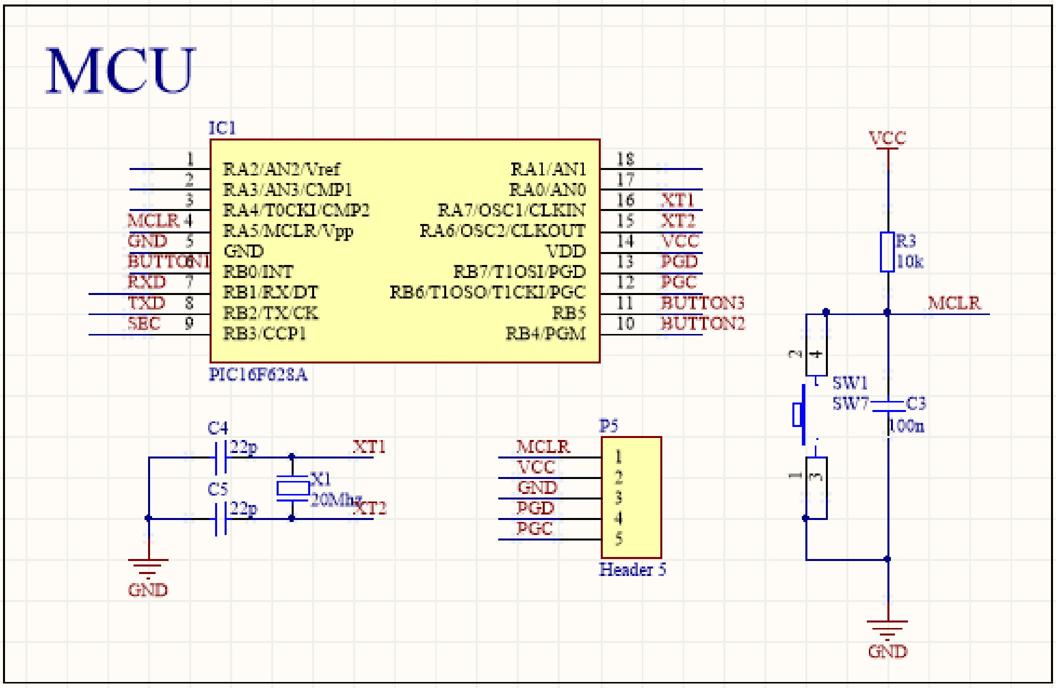

Figure 3.15: MCU of Slave Button(s)

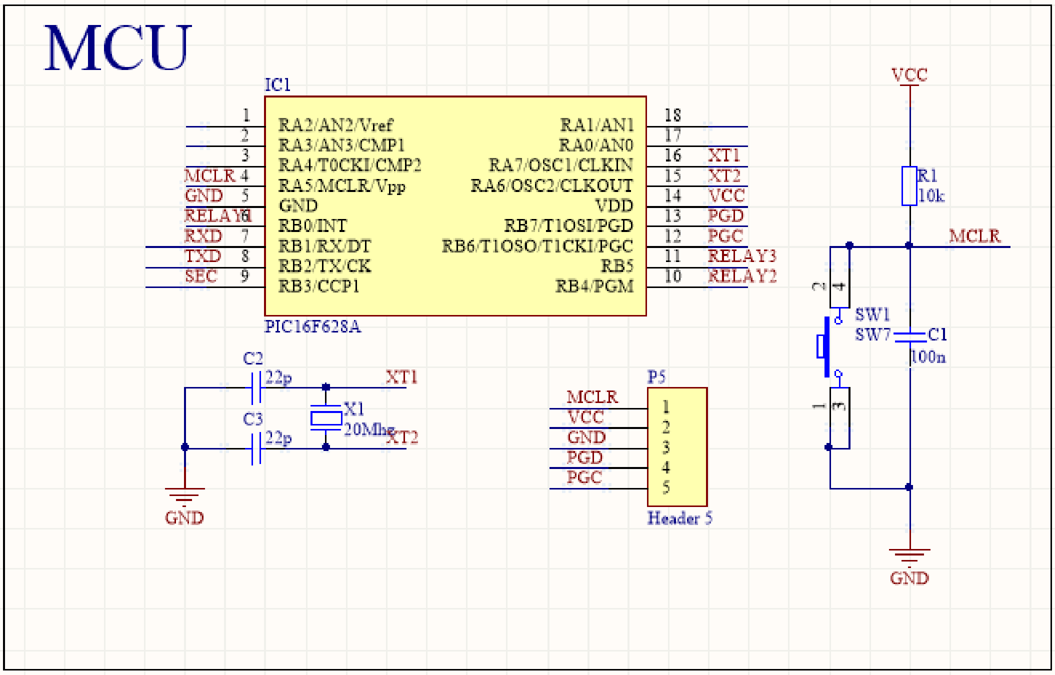

Figure 3.16: MCU of Slave Relay(s)

author used the same design for each button block, only increase or decrease the number of button if needed.

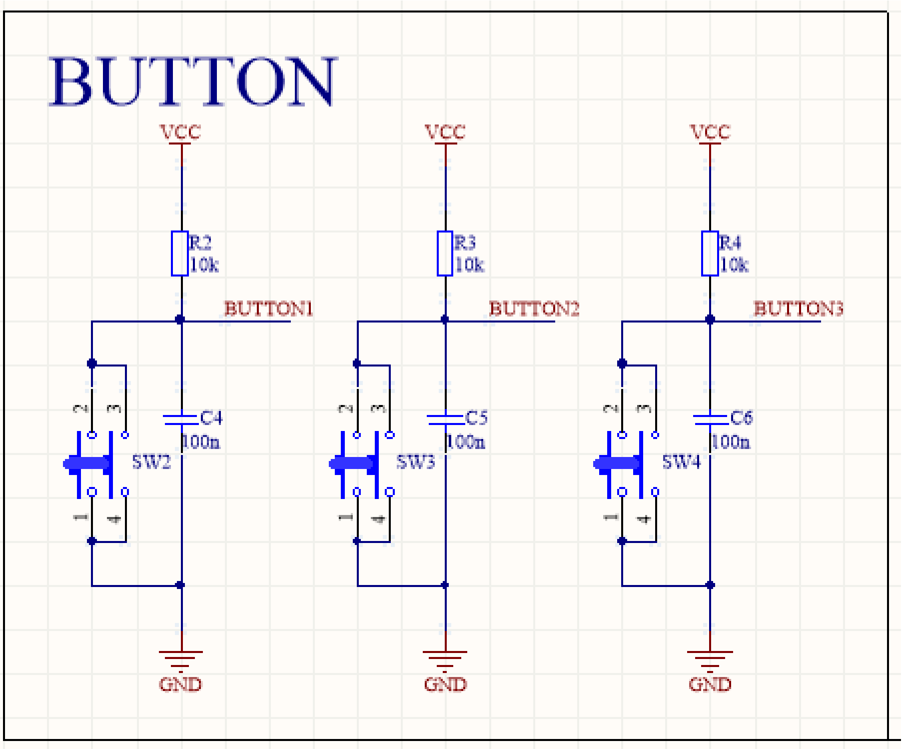

Figure 3.17: Button block of Slave Button(s)

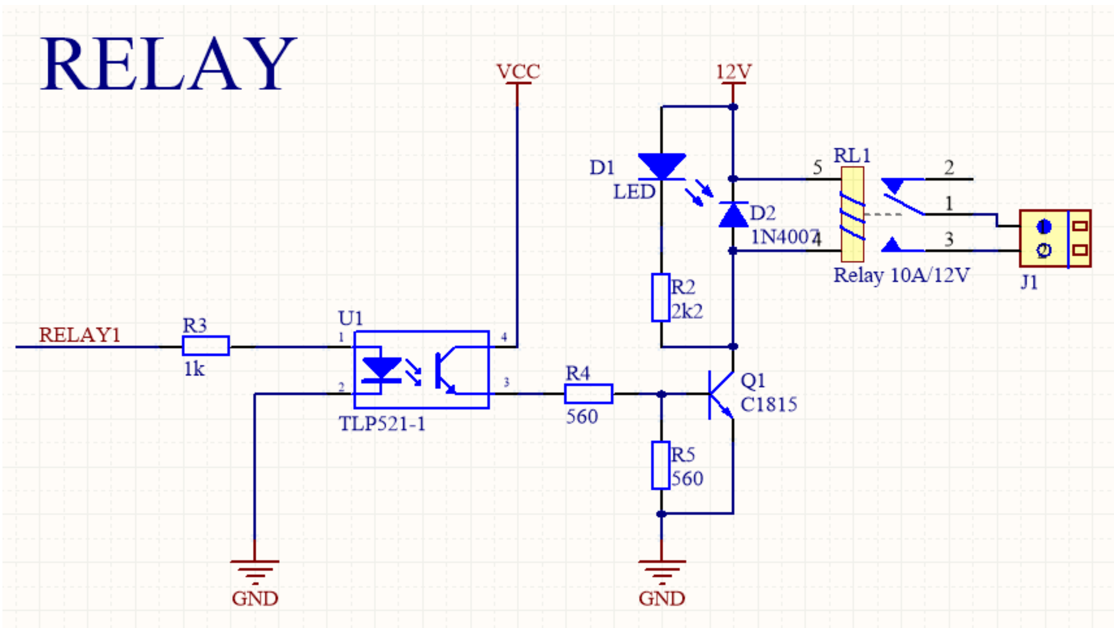

Figure 3.18: Relay block of Slave Relay(s)

3.3.6 Relay block of Slave Relay(s)

Relay is an electrical component which operates as a switch under electromagnetic working principle. It is useful when users need to switch state to control one to many circuit under one signal. A relay has two states are On and Off, switching bases on the current flows through its coil. Relay with parameters of 5 pins and 12V-10A DC is chosen in this thesis to ensure the switching circuit will operate accurately through long distance cable. Similar to Button block of Slave Button(s), the author use one design of a relay block and then increase or decrease the number of block in case needed. Figure 3.18 shows one relay block of one of the Slave Relay(s) in the project.

3.4 Security Camera Block

When mention a security camera a few years back, people only think about an expensive system that only company level may afford and the system is massive itself, which make its mobility extremely low. However, world is changing rapidly, and with a household level, people still can implement a security camera system without paying a huge price but still, be provided with acceptable quality and performance. Furthermore, end users not only can view the security camera in fixed place but also can view camera from anywhere with Internet connection, an embedded computer and a device can connect to the Internet. In this thesis, the author built two function for Security Camera Block, which are Facial Recognition and Motion Detection. The first function is embedded onto a Mini Computer named Raspberry Pi 3, please see Figure 3.19, and second function, which is limited by resources from the author, will be integrated directly into Application Server as a prototype only.

Figure 3.19: Raspberry Pi 3 Model B

Raspberry Pi 3 has market share at third place only followed by Mac and Windows. It is a mini computer, which means it has small size but has been integrated with every component that makes it a computer can in an ATM card size. Raspberry Pi 3 is the most powerful option of Pi series, but it has reasonable price with widely support documents and community because its operating system is Linux or Windows 10 IoT. Also, with the supports from the operating system itself and the specifications listed as following, it is a suitable computer for this project. Besides, instead of using an USB camera through USB port, which also compatible with other types of computers, the author used camera modules that attached directly onboard of Raspberry Pi through CSI as in Figure 3.20. The significant advantage of the camera module compare to other USB camera is the huge different of quality of camera input stream and processing speed because it is built for Raspberry Pi. Please see Table 3.6 for few highlight specifications of Pi Camera Module. After connecting Pi Camera Module with Raspberry Pi 3 through CSI port via ribbon cable, Raspberry Pi should have the ability to view, record video, connect with and control the system via Wi-Fi connection after some configurations in next chapter of this thesis. Figure 3.21 shows the standalone Raspberry Pi with Pi Camera Module attached. Furthermore, in the event that the owner need to add people into their recognition database, they can do it directly with Raspberry Pi and its PiCamera Module.

Raspberry Pi 3 Specifications:

- SoC: Broadcom BCM2837.
- CPU: 4 core ARM Cortex-A53, clock 1.2GHz.
- GPU: Broadcom VideoCore IV.
- RAM: 1GB, bus 900Mhz.
- Connection: 10/100 Ethernet, 2.4GHz 802.11n, Bluetooth 4.1 Classic, BLE.
- GPIO: 40 pin.
- Others: 4x USB port, microSD, camera module port, HDMI.

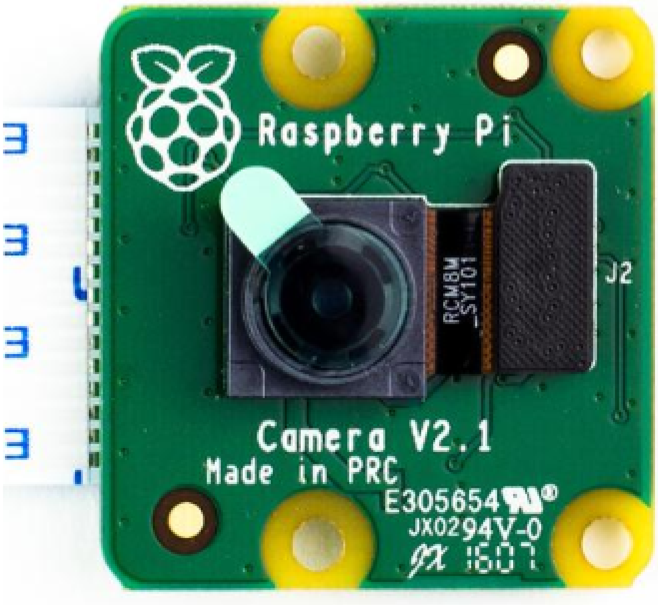

Figure 3.20: Raspberry Pi Camera Module

Attribute	Detail
Weight	3g
Resolution	8 Megapixels
Video modes	1080p30, 720p60 and 640 x 480p60,90
Linux integration	V4L2 driver available
Sensor resolution	3280 x 2464 pixels
Sensor image area	3.68 x 2.76 mm (4.6 mm diagonal)

Table 3.6: PiCamera Module Highlight Specifications

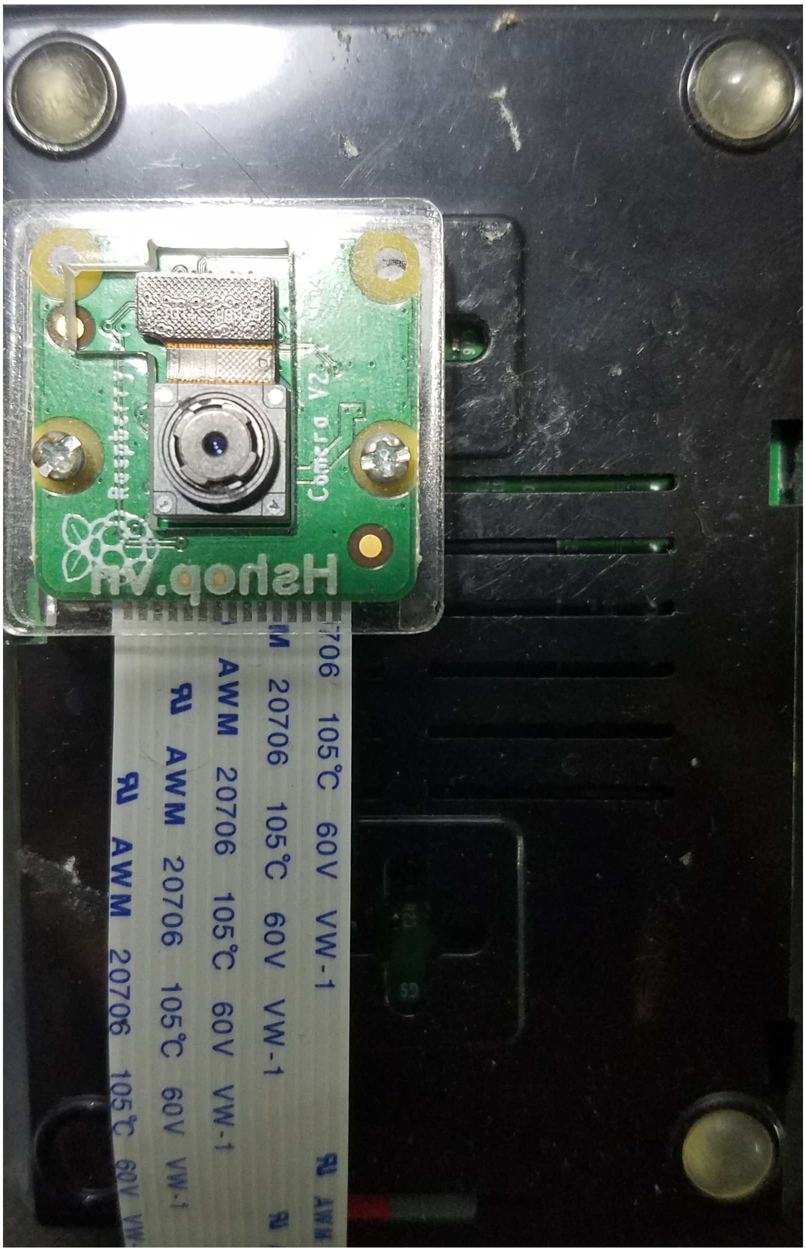

Figure 3.21: Raspberry Pi Connected with PiCamera Module

3.5 Full Schematic of the System

Following are the full schematics of the system, namely Power for Master, Master, Slave button(s) and Slave Relay(s), respectively. The other modules which are not mentioned in section 3.2 are the parts that may be extended in the future.

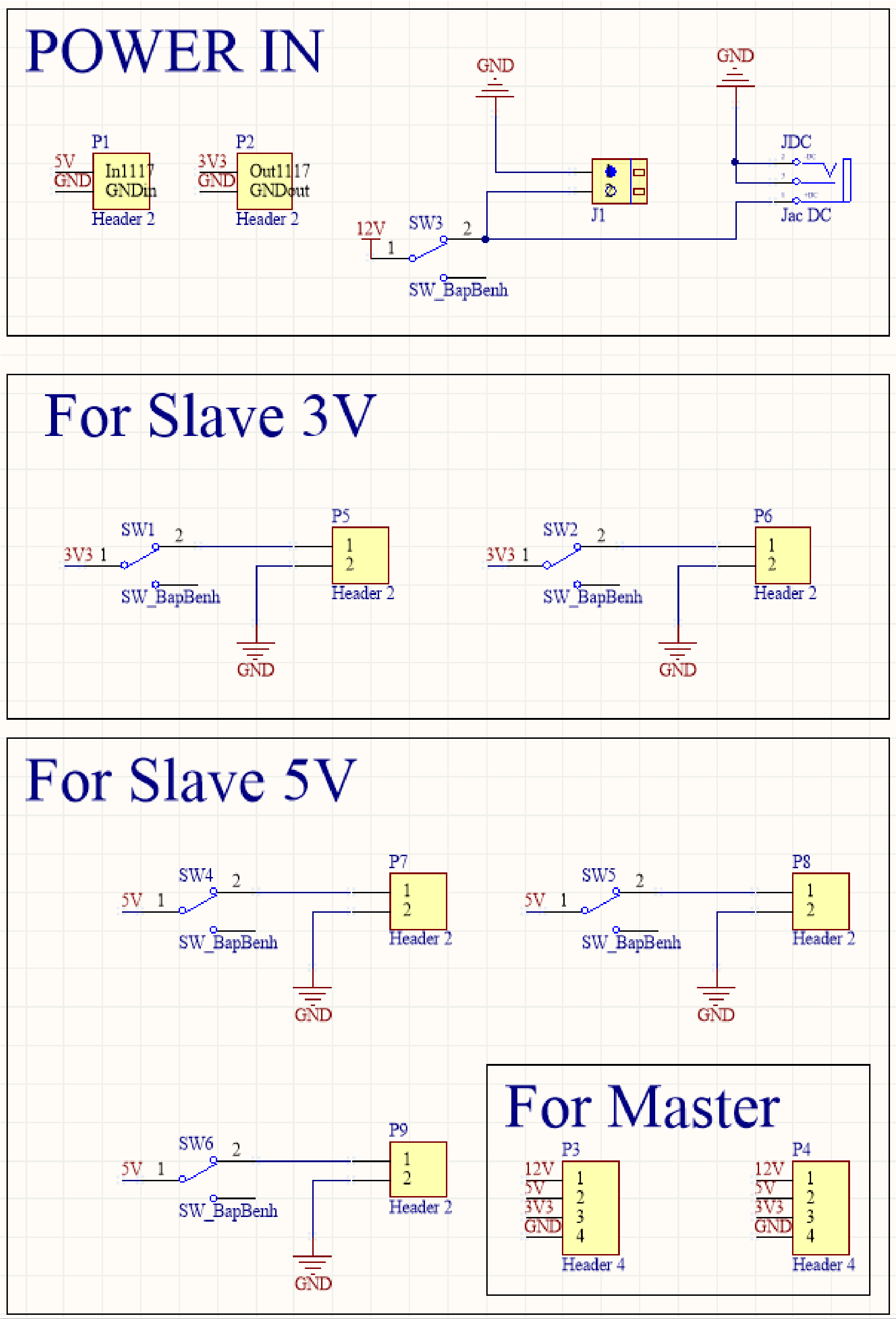

Figure 3.22: Full Schematic of Power for Master

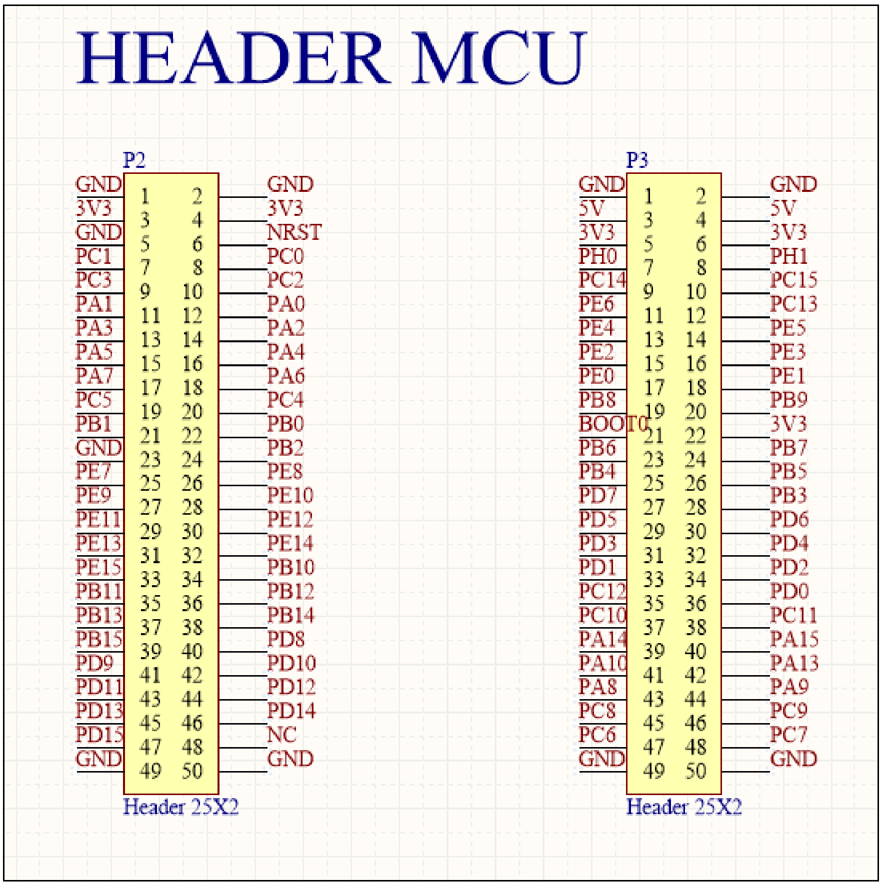

Figure 3.23: Header for STM32F4 Discovery Kit

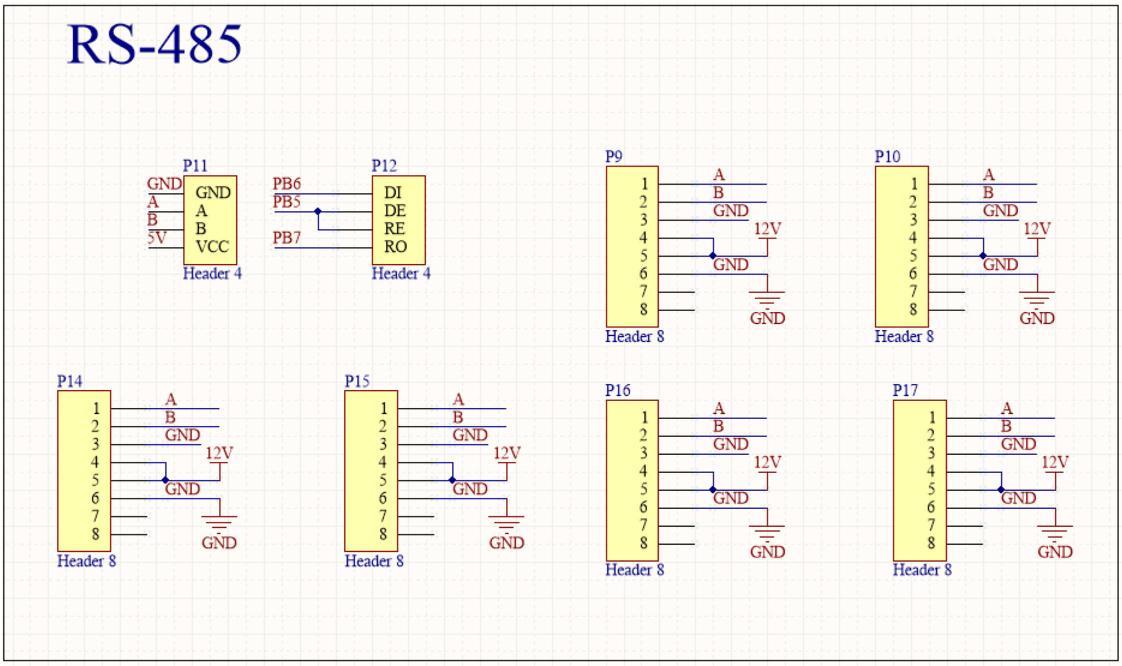

Figure 3.24: Master: Header for module RS-485 of Master

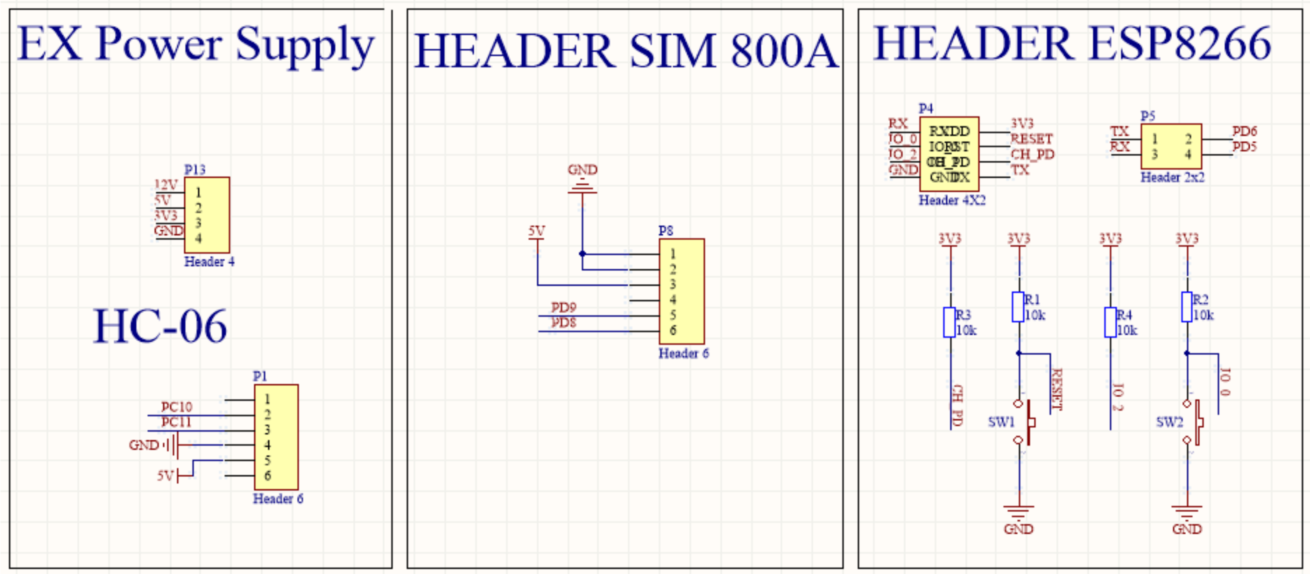

Figure 3.25: Master: Header for other modules

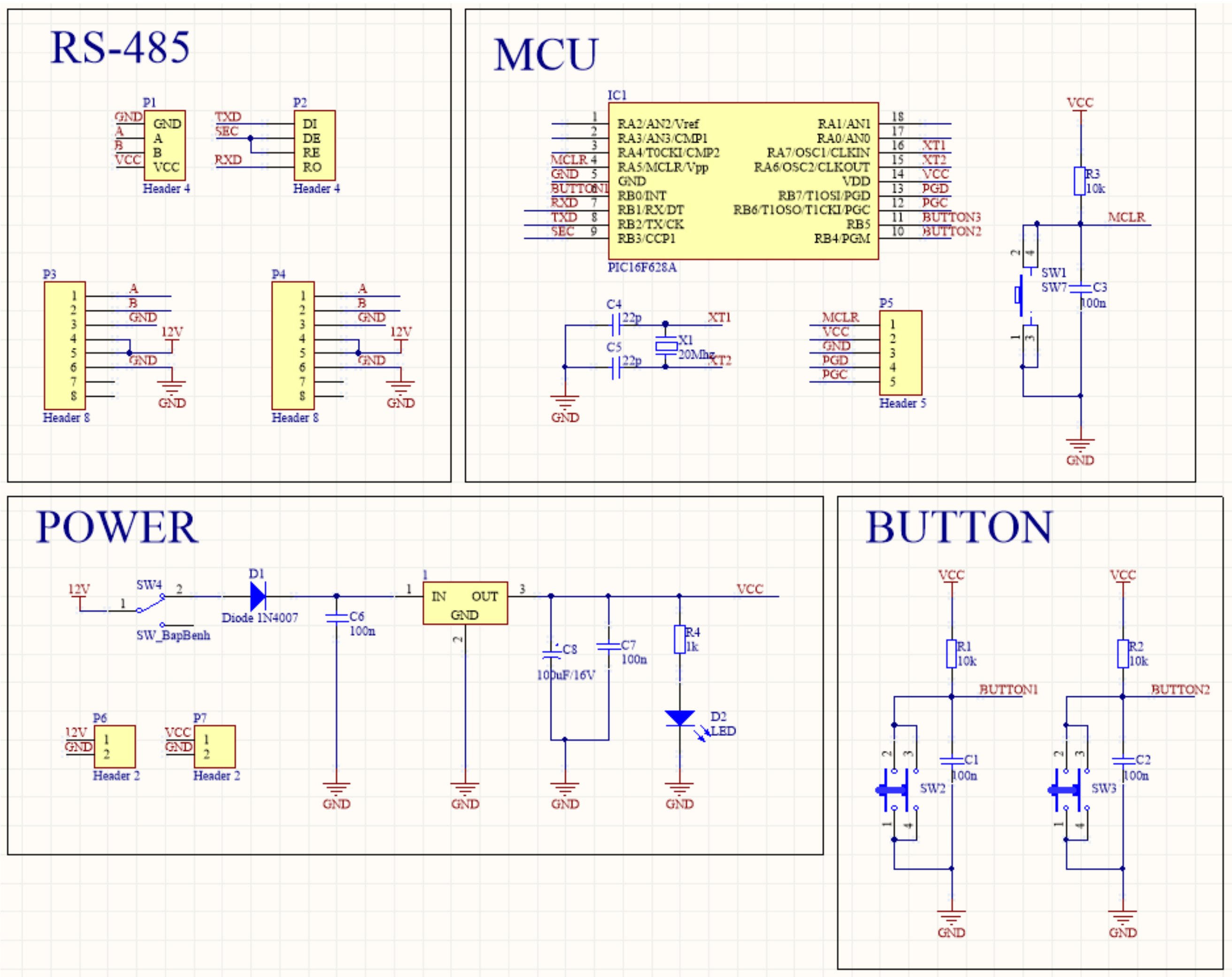

Figure 3.26: Slave Button(s)

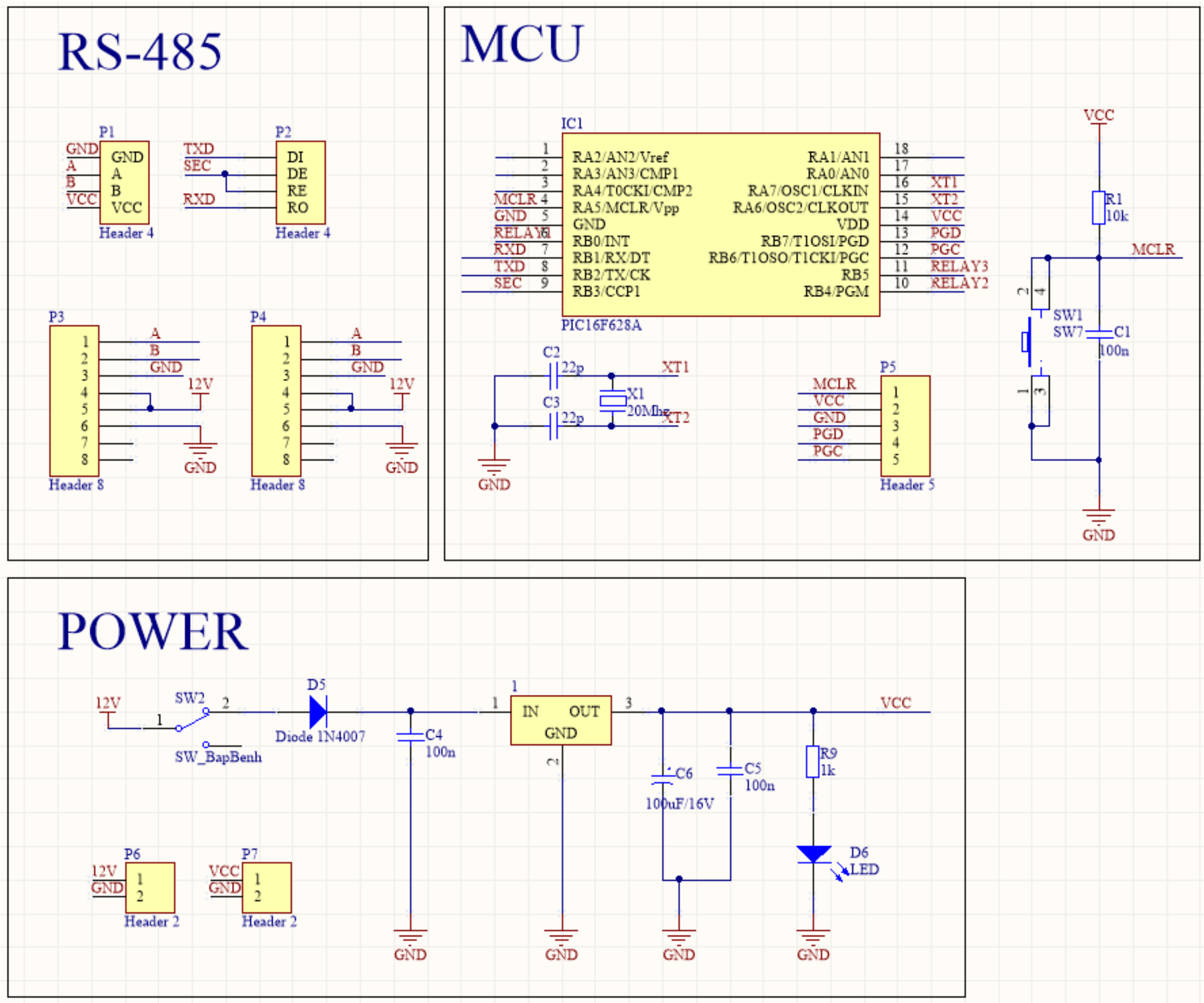

Figure 3.27: Slave Relay(1)

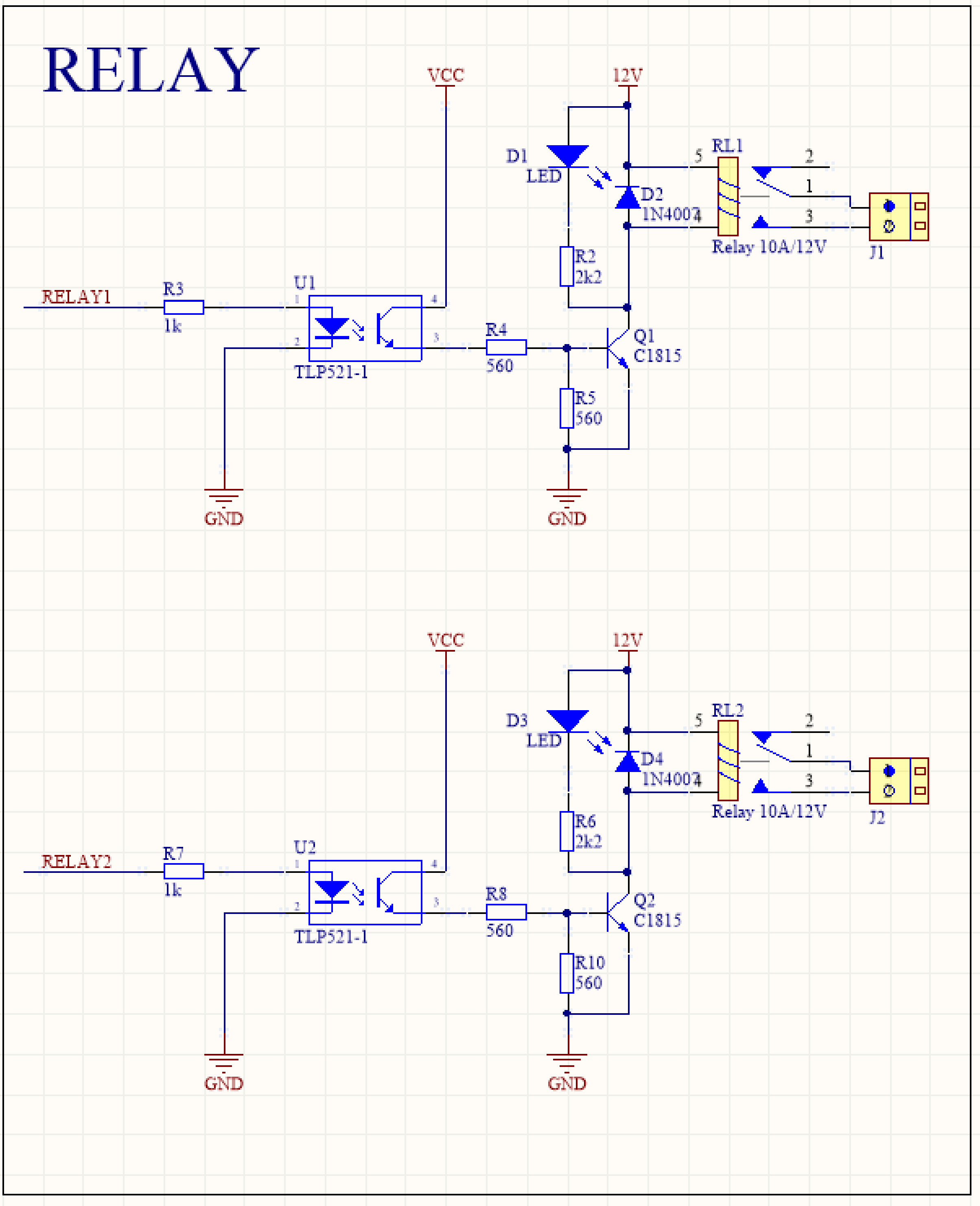

Figure 3.28: Slave Relay(2)

Chapter 4

Algorithm and Software Design of The System

4.1 Features explanation

In this thesis, the system must have the ability to control devices in the house in a convenient way, which means user can control devices from anywhere, anytime they want with an Internet-connected device and timer. Furthermore, the house should be capable of providing scenarios depends on the needs of user. For instance, users can switch on or off a number of specific devices by choosing a scenario instead of choosing individual device, namely “I'm home”, “Good night”, or “I'm leaving”. Based on the basic ideas, the thesis is integrated with all the basic features above. The thesis is designed for an one-floor house with three rooms, namely Living Room, Dining Room and Bedroom.

4.1.1 Convenient control

Living Room

Front part of the house, which includes Main Door, Living Room and a Stair, the belonging AC devices will be controlled with four Relays but integrated into two Slave-Relay(s), one is Slave-3-Relays and the other is Slave-2-Relays (which is also integrated with the Relay controlling the Conditioner of Bedroom).

Dining Room

Second part of the house is assumed to have only one AC device and will be controlled with a Relay integrated on a Slave-2-Relays, in which has the Relay to control the Bedroom Light.

Bedroom

Last room of the house is Bedroom, in which is assumed to have two devices but one is integrated on the same circuit with Dining Room Light, the other is the Conditioner is also integrated on the same circuit with the Living Room Light.

Apart from controlling the devices by physical Slave Button(s), which is also crossed implemented with no specific rule, the owner also can control the devices with a single Internet-connected device such as a smartphone, a tablet or a computer by accessing the Application Server from anywhere and anytime. Besides, it also has few scenarios that should be quite helpful for the owner. Imagine that when the owner arrive home after work, the devices needed are ready to serve such as the Front Light or the Conditioner. The project is also implemented with security camera block which helps user to access, monitor their house and receive alert in case of abnormal event happens in a convenient way with reasonable price. In addition, all data in the process of monitoring the house should be sent to a database, which helps the user and also engineer can keep track of the activities of devices in the house, then use the collected data to improve the experiences of the users in the future.

4.1.2 Block Diagram

Figure 4.1 refers the overview of the system. From the block diagram, there are three main blocks, namely Master, Slaves and Internet Application block. In this thesis, each main block has different functions and may consists of one to many smaller blocks. Referring to Figure 4.1, Master is in the middle, connects Slaves and Internet Application Block; Slaves are the "workers" depend on the Master and the Internet Block helps the User communicate with the system through Master remotely.

Master

As designed in section 3.2, Master is the circuit integrated with a STM32F4 Discovery Kit, connected via headers instead of being soldered directly on board in order to ensure an effortless replacement if broken. Beside the block of RS-485 module for main communication methodology and ESP-8266 for establishing connection to the Internet, it also has the headers for other modules of connectivity and functions in order to make the Master scalable in the future, namely SIM800A, Bluetooth module HC-05, and Real time module DS3231. However, instead of using an integrated Power block onto Master circuit, it uses a separated Power for Master as mentioned in section 3.2.5.

Based on the basic idea, Master is responsible for receiving the requests from all sources, Slaves or Internet Application block, and distributing the command to the Slave with appropriate function. In addition, Master is also the middleman between Internet Application block with the Slaves, which means it also update the information between Internet application block and Slaves.

Slave Relay(s)

Slave Relay(s) (Slave #1, #2, #3) consists of number of Relays (varies depends on users' needs) and one PIC16F628A from Microchip as the MCU, responsible for switching AC devices On or Off based on the distributed command from Master.

Slave Button(s)

Slave Button(s) (Slave #4-8) consists of number of Buttons (varies depends on users' needs) and one PIC16F628A from Microchip as the MCU, responsible for

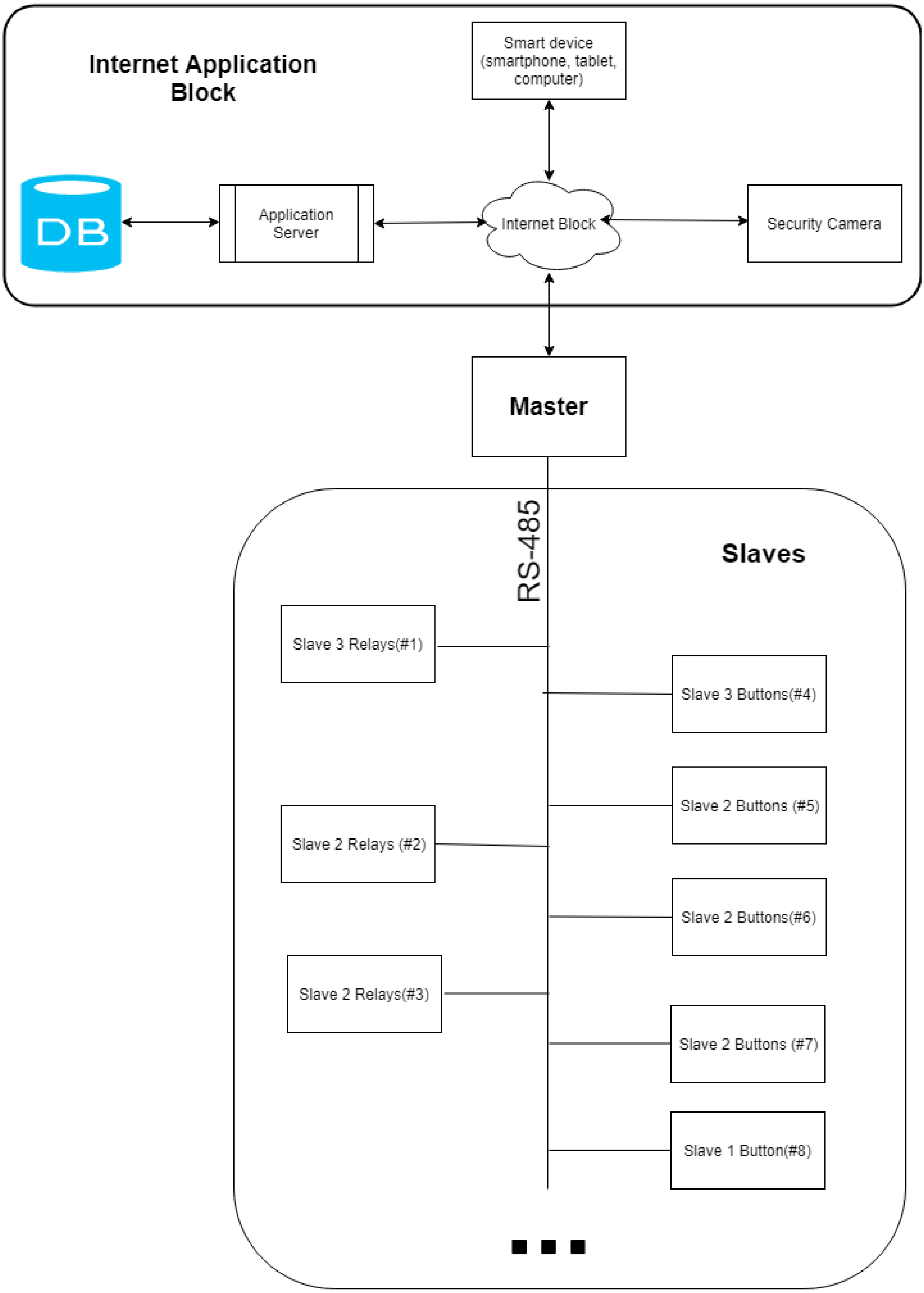

Figure 4.1: System Block Diagram

controlling Slave Relay(s) by sending the commands to Master for the distribution to corresponding Slave Relay(s).

Internet Application Block

Internet Application block consists of smaller blocks with different functions, namely Database, Application Server, Internet connection block, Security Camera and Smart devices. Internet block establish the Internet connection for Master block; Application Server is the combination of back-end and front-end development of Webserver, besides helping users to control the system remotely with ease, the extended features will be explained later in this chapter; Security Camera responsible for recognizing person with Facial Recognition to open the door in order to cut off the steps of accessing the house. In addition, security camera is also integrated with motion detector prototype directly in Application Server.

4.1.3 Communication Methodology and Algorithm

In this thesis, data is transmitted from UART of MCU to input of module RS-485 then to the data bus wire to distribute to corresponded components, noted that RS-485 is the physical standard which helps transmitted data travels much further compared to original UART. It needs two wires for data transmitting with module RS-485, but a cable of four wires is chosen for providing power supply of 12VDC and transmitting data at the same time with a single cable. Four wires in a cable with corresponded functions are listed as following.

- 12V: provide 12VDC throughout the system.
- A: Signal wire A.
- B: Signal wire B.
- GND: Common ground throughout the system.

However, RS-485 is a physical standard instead of an algorithm to distribute data through the whole network of a large number of devices with acceptable performance. Furthermore, in order for the chosen module RS-485 be able to work, its enable pin must be controlled by the MCU, which is set to logic 0 as default is receiving mode and vice versa. After sometimes reading books, the author suggested two algorithms for this thesis.

Ask/Request sequentially

- **Master**: Master responsible for asking sequentially every connected Slaves in the system. After asking the Slaves for if they need to work, Master will delay for a small amount of time to wait for the response from Slaves. If the time is passed and asked Slave does not request to work, Master will pass that Slave and move on the next Slave. If a Slave Button is being asked but also receive the external signal, it can interrupt the process by sending a response to Master requesting to work, then that request will be prior to be sent to corresponded Slave Device. Now it ends the loop and start a new one. Period of time to ask the Slaves must be in milliseconds in order to complete the loop for every Slaves in the system.

- **Slave Button**: Slave Button has to wait for Master to ask and response. It is always stay in receiving mode (which has enable pin logic at 0 – LOW). When Slave Button is asked or an external signal comes in, it pulls enable pin to logic 1-HIGH to enter transmitting mode and response when Master asks.

- **SLave Device**: Enable pin of RS-485 of Slave Device is always in LOW status, which means receiving data mode. When Master ask or there is data transmitted from Slave Button, Slave Device will check if it is corresponded with its functions, if yes Slave Device will work as defined function.

With this algorithm, the asking loops will run continuously, and it should prevent two signal collide with one and another because the Slave only answers Master when Master asks. However, transmitted data will be difficult to be managed because of two reasons, waiting time and management. After a request is sent from Master, it takes some time for Master to wait for the response from Slave and pull enable pin of RS-485 to LOW. Besides, it takes times again to pull enable pin up to logic HIGH to distribute the response if available. This process is getting longer with the increment of the number of Slaves, which cause the transmission between Slaves become slower with a large number of Slaves. Furthermore, transmitting data continuously will consume loads of bandwidth leads to resource waste and cause errors dues to noise or inaccurate process from Slaves because of data is transmitted continuously leads to false data or worse is lose data. Thus, the author chose a different method to transmit data through the network based on CSMA/CD protocol.

Chapter 5

Experimental Results

Chapter 6

Conclusion