Raspberry Pi

Hardware View

ELECTGON
www.electgon.com
ma_ext@gmx.net

26.04.2018

Contents

Abstract

In less than five years, a tiny computer board has become one of the best seller computer all over the world. Its production started in 2011 aiming to have this credit-card size board as a learning platform for pupils who are interested in learning programming. But its usage started to expand in many fields also in the industry. Therefore, this document provides introduction and Hardware view of Raspberry Pi to have an overview of how this board can be used.

1 Introduction

It is common now to meet many people who are using Raspberry Pi board in their houses to utilize some functions like weather reporting or online radio listening. It became popular also to have summer school for children intention learning programming. Not only but also many applications in the industry started to use Raspberry Pi for performing specific functions in Control and Embedded Systems development.

This widely used board is using ARM processor as its processing unit.This is one important point for making it popular. Also availability of its resources, SW packages and complementary peripherals strengthen the trend to use it. Because of that, it is useful to have good knowledge about this board, understand its components and architecture as a brief to start having some applications using this board.

Figure 1: Raspberry Pi Board[1]

Figure 1 shows a Raspberry Pi board. All what you need to make it work is to connect a power supply, Monitor, Mouse and Keyboard. In some cases, you don't need any peripheral to make it work you need then only a power supply and Ethernet connection.

2 Raspberry Pi Versions

Since 2011, Raspberry Pi is produced with more improvements are applied by the time. So far (until time of writing this article) we can say that there are mainly three generations of the Raspberry Pi. Within each generation, you can find different models are produced. For example in first generation (Raspberry Pi 1) you can find models B, A, B+ and A+. Second generation (Raspberry Pi 2) was produced with only Model B. Raspberry Pi 3 has 2 models B and B+. Figure2 shows these different boards.

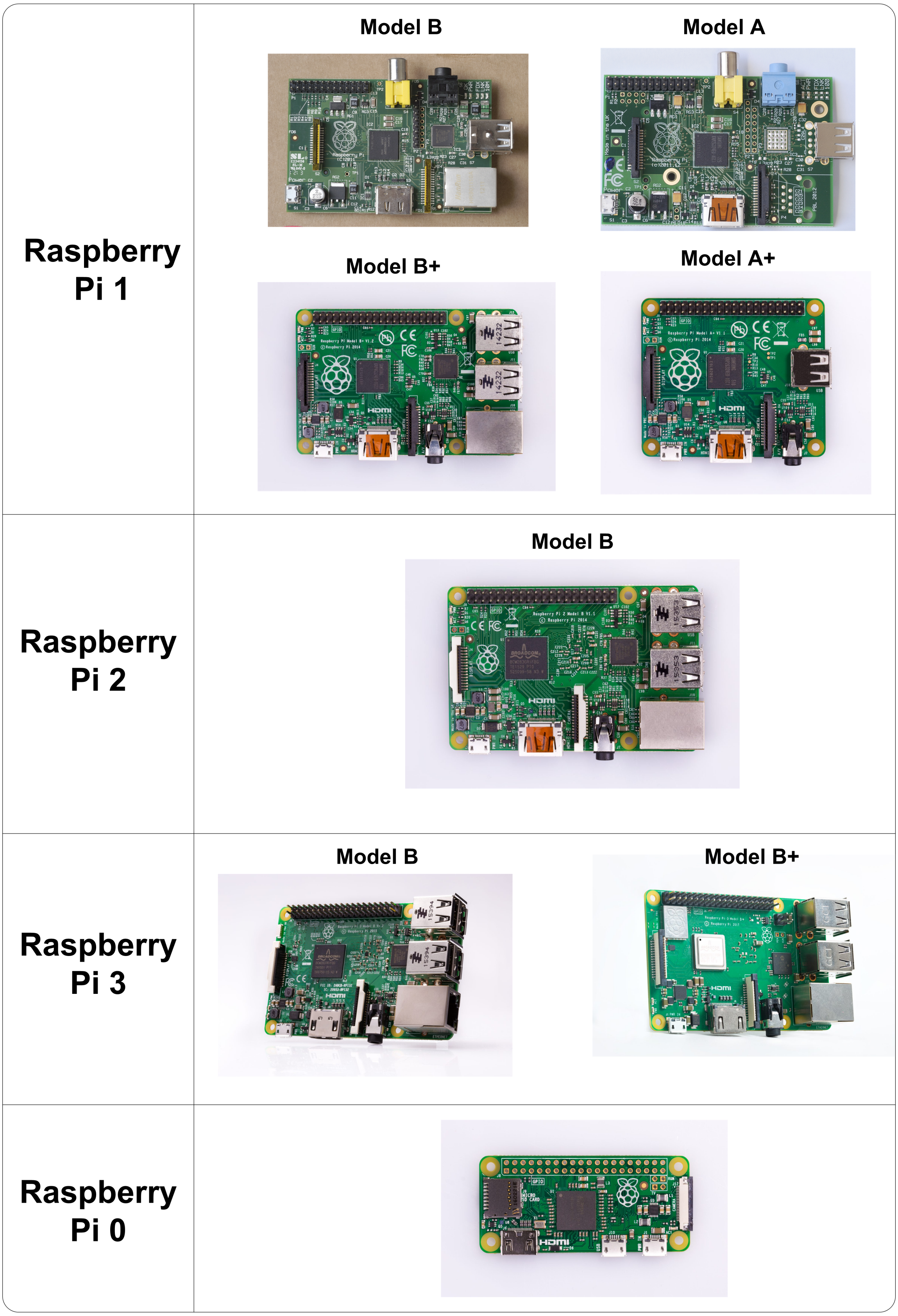

Figure 2: Raspberry Pi Models[1]

There is no common features across Models A nor B. But as some improvements are applied, New model is produced to provide new improvements. Table 1 shows basic comparison between features of these models

<table>
<tr><th></th><th>Pi 1 B</th><th>Pi 1 A</th><th>Compute Module</th><th>Pi 1 B+</th><th>Pi 1 A+</th><th>Pi 2 B</th><th>Zero</th><th>Pi 3 B</th><th>Pi 3 B+</th></tr>
<tr><td>Release</td><td>Apr 2012</td><td>Feb 2013</td><td>Apr 2014</td><td>Jul 2014</td><td>Nov 2014</td><td>Feb 2015</td><td>Nov 2015</td><td>Feb 2016</td><td>Feb 2018</td></tr>
<tr><td>RAM</td><td>512 MB</td><td>256 MB</td><td>512 MB</td><td>512 MB</td><td>512 MB</td><td>1 GB</td><td>512 MB</td><td colspan="2">1 GB</td></tr>
<tr><td>System on Chip</td><td colspan="5">BCM2835</td><td>BCM2836</td><td>BCM2835</td><td>BCM2837</td><td>BCM2837B0</td></tr>
<tr><td>Processor</td><td colspan="5">ARMv6</td><td>ARMv7</td><td>ARMv6</td><td colspan="2">ARMv8</td></tr>
<tr><td>Cores</td><td colspan="5">1</td><td>4</td><td>1</td><td colspan="2">4</td></tr>
<tr><td>Clock</td><td colspan="5">700 MHz</td><td>900 MHz</td><td>1 GHz</td><td>1.2 GHz</td><td>1.4 GHz</td></tr>
<tr><td>USB 2.0</td><td>2</td><td>1</td><td>1</td><td>4</td><td>1</td><td>4</td><td>1</td><td colspan="2">4</td></tr>
<tr><td>Ethernet</td><td>10/100 Mbps</td><td>None</td><td>10/100 Mbps</td><td>10/100 Mbps</td><td>None</td><td>10/100 Mbps</td><td>None</td><td>10/100 Mbps</td><td>1 Gbps</td></tr>
<tr><td>WiFi</td><td colspan="7">No</td><td colspan="2">Yes</td></tr>
<tr><td>Bluetooth</td><td colspan="7">No</td><td colspan="2">Yes</td></tr>
<tr><td>Storage</td><td colspan="2">SD card</td><td>eMMC flash</td><td colspan="6">Micro SD card</td></tr>
<tr><td>GPIO</td><td colspan="2">26 Pins</td><td>54 Pins</td><td colspan="6">40 Pins</td></tr>
</table>

Table 1: Raspberry Pi Features Summary

3 Operating Systems

Once you have your Raspberry Pi board is ready and connected with needed peripherals, you can start working with it as long you have an operating system is installed in its attached micro SD card. Available operating system images for Raspberry Pi can be found in its website

https://www.raspberrypi.org/downloads/

On this page you can find ready images like Ubuntu mate, Ubuntu Snappy, Windows 10 and Raspbian. This Raspbian is built to optimized for Raspberry Pi. This operating system (or root file system to be more specific) is built based on Linux kernel which enables user configurations to be applied (like realtime patch for instance).

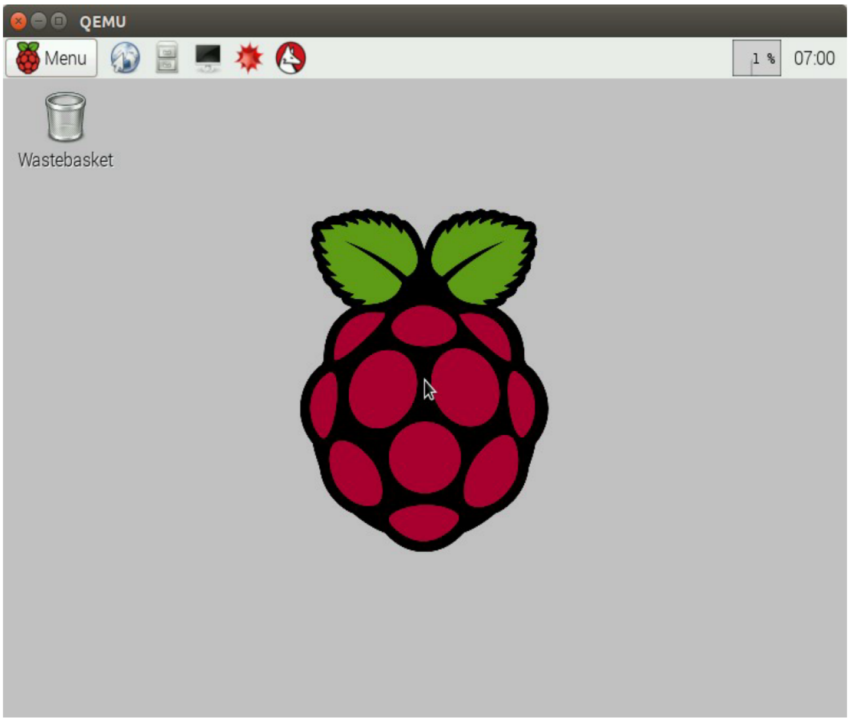

Figure 3: Raspbian Operating System

4 Specifications of Raspberry Pi

To use Raspberry Pi in your system, it means a kind of integration of processing unit in your system. So it is important to understand how to add this unit into your system. Now it is turn to discuss slightly more details about Hardware structure of the Raspberry Pi. Later we can see how to build your specific application in Raspberry Pi.

As a processing unit, Raspberry Pi is built on ARM processor along with important interfacing that enables developers to exploit Raspberry Pi best in their projects. Old versions of Raspberry Pi used ARMv6 architecture. Raspberry Pi2 started to use ARMv7 architecture which is a 32 bit processor architecture. Raspberry Pi3 is using ARMv8 architecture which is 64 bit processor architecture. This architecture is produced by Broadcom chip BCM2837.

What is important then to know about this board is which interfaces are included in Raspberry Pi board like Ethernet port, 4 USB ports, HDMI port..all of these interfaces are important to work with Raspberry Pi as a standalone computer system. We will take Raspberry Pi 3 in more details for instance here to get familiar with some important hardware specifications as shown in table 2

Model Name	Raspberry Pi3 - B
System on Chip	Broadcom BCM28367
CPU	1.2 GHz, Quad Core, 64 bit, ARM Cortex A53
GPU	Broadcom VideoCore IV, OpenGL ES 2.0,OpenVG 1080p60 H.264 high-profile encode/decode, 400 MHz
Memory (SDRAM)	1024 MB
USB 2.0 ports	4 (via integrated USB hub in LAN9514)
Video outputs	HDMI
Audio outputs	- TRS connector (3.5 mm jack) - HDMI
Onboard Storage	MicroSD slot
Onboard Network	- 10/100 wired Ethernet RJ45 - Integrated 802.11n Wi-Fi & Bluetooth 4.1
Power source	- 5 V (DC) via Micro USB type B or - GPIO header - Current from 800 mA up to 2.5 A
Size	- 85.6 x 56.5 mm x 17mm - Weight 45 gm

Table 2: Raspberry Pi 3 Specifications

To simplify some terms of these specifications and to shed some light on important notes, what is meant by the field System on Chip is an IC provided by Broadcom that contains the processing unit. As mentioned earlier, each generation of Raspberry Pi has its improved processing unit (CPU). For the case of Raspberry Pi 3, its CPU is ARMv8 (cortex-a53), which is 64 bit processor with 1.2 GHz max. operating frequency.

GPU is another processing unit attached to the CPU and responsible for processing video applications. It is important to note about USB interface as it is using LAN9512 chip which is a USB Hub as well as Ethernet controller. This means that the Ethernet interface and USB ports are connected to this chip. This is important to note for applications that are based on Ethernet connection for examples and consumes much power which may cause degradation in Raspberry Pi performance in some cases. Providing the appropriate power then is important to Raspberry Pi board. Although it is possible to provide power through GPIO pins, but for stable performance it is advised to use external power via MicroUSB port and make sure that the power supply can provide around 2 A to the board.

As a new feature in Raspberry Pi 3, there is integrated Wi-Fi and Bluetooth adapter in the board. Which means you can utilize the board for wireless connection without the need to buy external dongle for wireless connection. Figure 4 shows diagram of Raspberry Pi 3 board and its components.

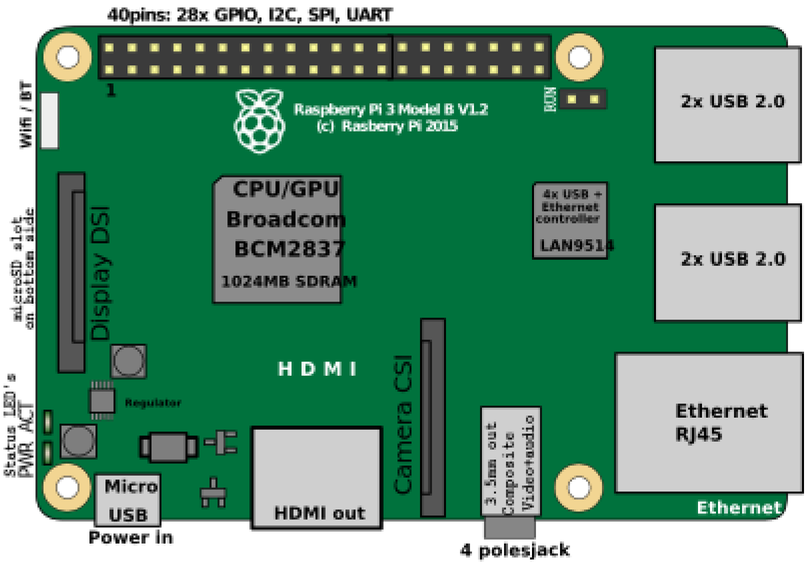

Figure 4: Raspberry Pi 3 Schematic [3]

Another important interface is GPIO interface which is general pins used for using Raspberry Pi integrated in your system. Figure 5 summarizes all GPIO pins used in Raspberry Pi.

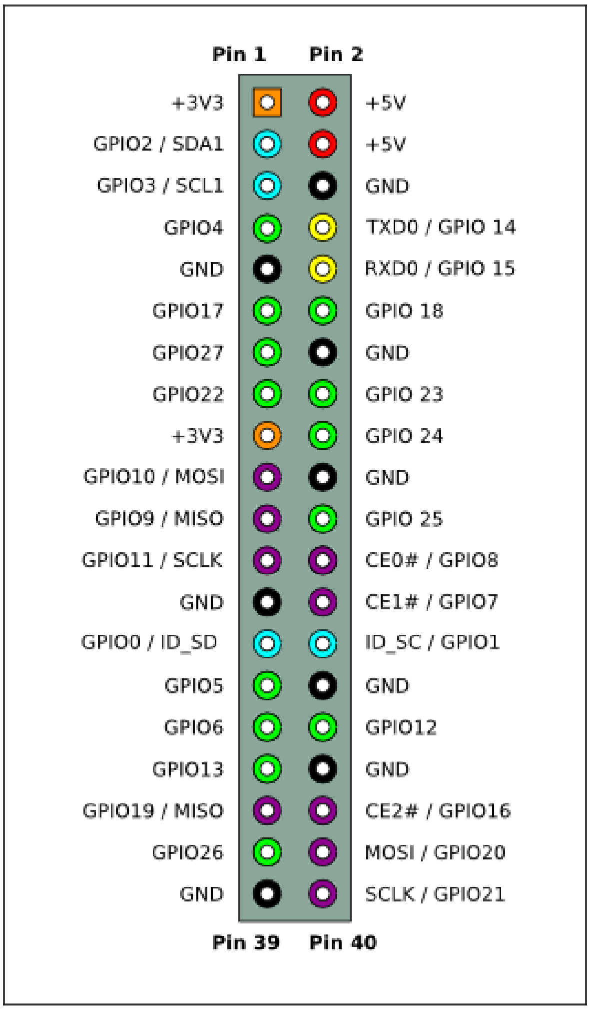

Figure 5: GPIO in Raspberry Pi[4]

In total, there are 40 GPIO pins. Each pin belongs to specific communication system. Raspberry Pi supports the following communication systems: UART. SPI, I2C. For example, pins number 4, 6, 8, 10 are used for UART connections. Pins numbers 11, 12, 19, 21, 23, 24, 26, 35, 36, 38, 40 are used for SPI connections.

To work with these pins configurations, there are ready libraries that can be used to ease access and controlling these pins. Communication over these pins are built using many programming languages so that you have variety of choosing your preferred programming language or according to your project requirements.

5 Expansion Boards

Working with GPIO of Raspberry Pi may be confusing and complex, therefore another easier level of integration maybe needed. For that purpose, some ready boards were developed in order to raise level of integration from this GPIO pins to signal level communication. This means that you can think about connecting your analog or digital signal to the Raspberry Pi without thinking about connection model and configuration with Raspberry Pi GPIO. That is because these expansion boards provide ready ports for connection with your system signals.

So as a developer you can integrate Raspberry Pi easily in your system. Figure 6 shows what is meant by expansion board and series of expansion boards can be stacked so that each board provides easier special interface.

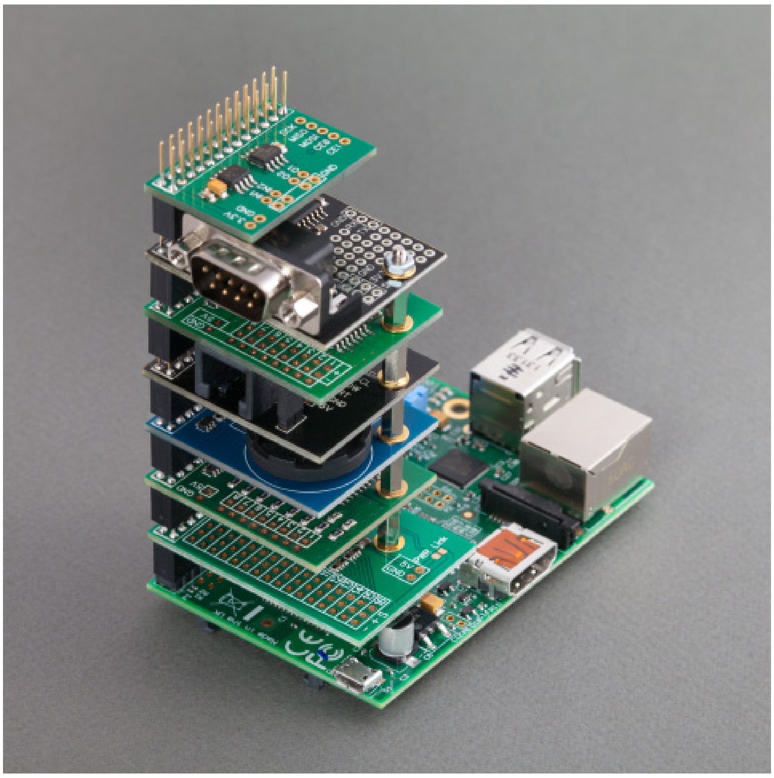

Figure 6: Stacked Expansion Boards [5]

Many expansion boards are available, each is serving specific applications or for general purpose. A list of these expansion boards can be found in this link

http://elinux.org/RPi_Expansion_Boards

6 PiFace Board

One important board is the PiFace board which allows your Raspberry Pi to control input and output devices in the real world . It has 2 changeover relays, 4 switches, 8 digital inputs, 8 open- collector outputs, 8 LED indicators and can be programmed in Python, Scratch and C. Figure shows main layout of PiFace board

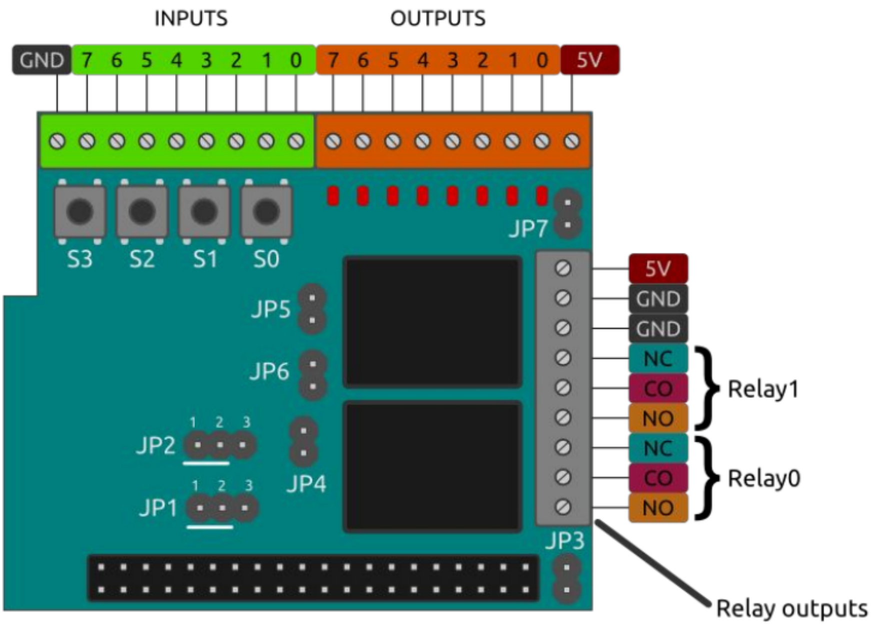

Figure 7: PiFace Layout [6]

Figure 7 shows main layout of PiFace which consists of 16 screw terminals. Eight of them or receiving digital input signal, the other eight for transmitting digital output signals. Relays used in this board are mechanical relays to enable applications that needs some switching mechanisms. To understand how these digital interface can be used, this can be depicted by connecting a switch to the input interface as shown in figure 8 in which you connect one terminal of the switch to ground, the other terminal to the high logic level (5 V for example). Pressing the button will trigger logic '1'.

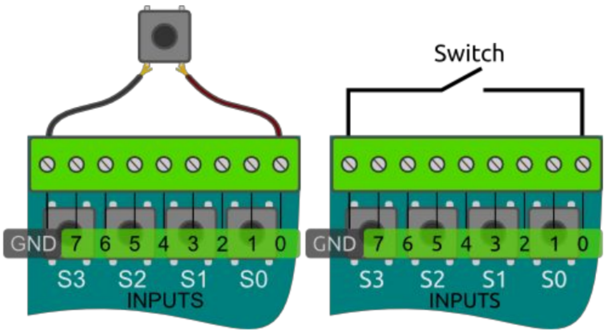

Figure 8: PiFace Input Example [6]

Simple example can be explained also using simple led as shown in figure 9

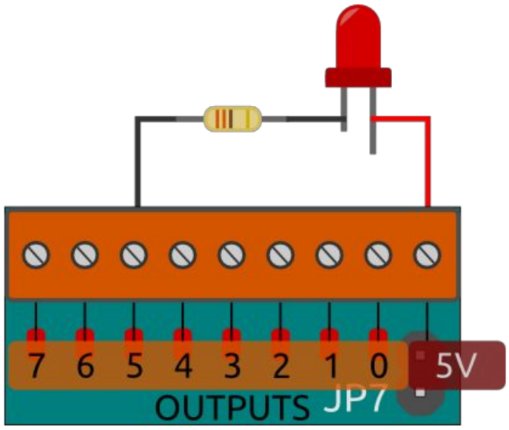

Figure 9: PiFace Output Example [6]

PiFace requires 3.3 V to operate. This 3.3 V can be provided from the Raspberry Pi. While the relays and output ports requires 5 V to operate which can be supplied from Raspberry Pi or externally.

6.1 PiFace Configuration

PiFace is communicating with the Raspberry Pi using SPI pins, so this should be enabled in Raspberry Pi configuration file. To enable it, open configuration file using the following command

```
$sudo raspi-config
```

When Configuration wizard opens, In Advanced Options menu select to enable SPI interface.

As an option, you can install Piface emulator SW which enables you to test and explore PiFace performance

To install the PiFace Digital 2 software, open a terminal and run the following command:

```
$sudo apt-get install python3-pifacedigitalio
```

To install the PiFace Digital 2 Emulator software, open a terminal and run the following command:

```
$sudo apt-get install python3-pifacedigital-emulator
```

6.2 PiFace Programming in C

To work with PiFace in C, many developers have developed libraries to provide C-based functions that can be used in C files. So that users who are familiar with C programming can utilize PiFace easily.

Two libraries were found to enable programming PiFace in C. First is libpifacedigital can be found at

http://piface.github.io/libpifacedigital/

The second C library was developed by Thomas Macpherson, one of PiFace developers. It can be found at his github space. To work with it, follow the following steps

Download the following C libraries in order to use its functions for commanding the PiFace

```
$git clone https://github.com/thomasmacpherson/piface.git
```

this will download directory 'piface', to install the library follow the following commands

```
$cd piface/c/
$./autogen.sh
$./configure
$make
$make install
```

This library should be installed then, you can start learning how to work with this library by reviewing sources folder this folder contains main functions that enables you working with the library. It can be found in INSTALLATION_DIRECTORY/c/src/piface

Open pfio.h to view available functions.

The following are examples that can guide you to understand how to program PiFace using C

Reading Input

```
#include <libpiface-1.0/pfio.h>
#include <stdio.h>

int main(void) {

//The following condition to check if board has initialized successfully or not
//pfio.init() returns -1 if there failure in initialization
 if (pfio_init() < 0)
            exit(-1);
 while (1) {
         printf("Input port: 0x%x\n", pfio_read_input());
        //pfio_read_input() gives value of all input ports
        sleep(1);
        //waits 1 second between two successive reading
        }
pfio_deinit();           //deinitialize the board

return 0;
}
```

Writing output

The following code makes flashing for output number 7

```
#include<libpiface-1.0/pfio.h>
#include<stdio.h>

int main (void)  {
 pfio_init();
```

```
 while (1)  {
     pfio_digital_write(7, 1);
     sleep(2);
     pfio_digital_write(7, 0);
 }

pfio_deinit();

}
```

Bibliography

[1] https://www.raspberrypi.org/

[2] https://www.networkworld.com/article/3111816/computers/the-discerning-nerds-guide-to-raspberry-pi-hardware-2016-mid-year-edition.html

[3] https://commons.wikimedia.org/w/index.php?curid=53283191

[4] https://elinux.org/RPi_Low-level_peripherals

[5] https://www.pinterest.de

[6] http://www.piface.org.uk/assets/docs/PiFace-Digital2_getting-started.pdf