Raspberry Pi - Morse Code Reader
Hardware Specifications

Author: Yashiv Fakir

Date: 20 January 2020

1 Introduction

This document aims to outline the hardware components necessary to run the Morse code reader. The document will begin with the components, then go on to the setup and how they differ for the various message inputs.

2 Part List

The parts in this list are not the most ideal components for this task and thus better alternatives do exist. For example, the ADC chip only needs one channel for this project where as the MCP3008 has 6 channels. Also a more sensitive LDR could be used to improve the measured voltage signal resulting in a better read of the encoded message.

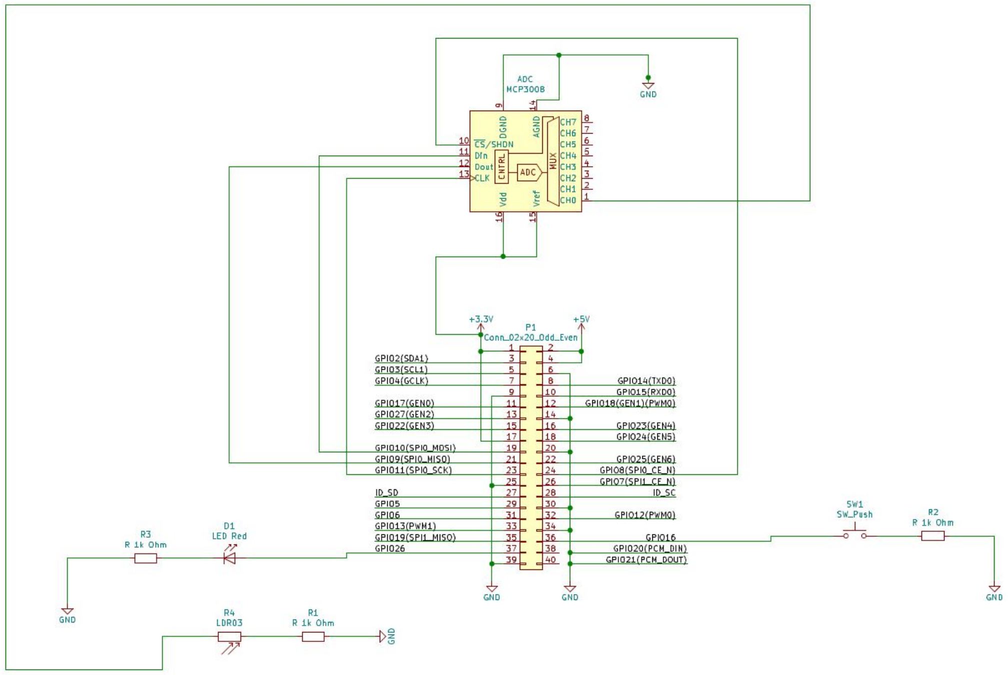
1. ADC Chip MCP3008 x1
2. 5mm Through-Hole LDR x1
3. 1 [kΩ] Resistor x4
4. Push-Button x1
5. Through-Hole LED (Red) x1
6. Through-Hole LED (Green) x1
7. Raspberry Pi (Pi Zero was used for this project)
8. Breadboard x1

3 Circuit Schematic

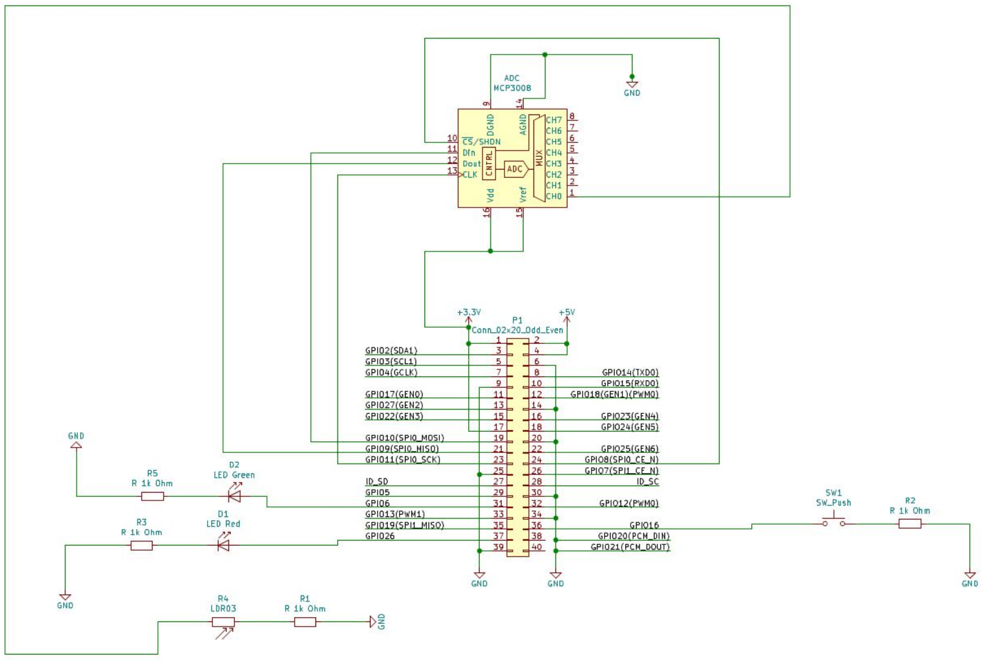
For both schematics it is very important that the LED/LED's and the LDR are positioned within close proximity to one another on the breadboard. This will decrease the environmental noise and improve the sensing ability of the LDR to pick up the change in colour on the paper encoded message or the light emitted from the LED encoded message.

Both schematics were produced using KiCad EDA, version 5.1.91. Refer to the reference section for download link [1]

3.1 Circuit Diagram for Paper Encoded Input



3.2 Circuit Diagram for LED Encoded Input



3.3 Technical Specifications

3.4 ADC

The MCP3008 ADC chip is linked to the Raspberry Pi using the SPI (Serial Peripheral Interface) protocol.

The exact connection ports between the ADC chip and the Raspberry Pi are as follows:

- CLK - GPIO 11 (SPI0SCLK)
- DOUT - GPIO 9 (SPI0MISO)
- DIN - GPIO 10 (SPI0MOSI)
- CS - GPIO 8 (SPI0CE0)
- VDD and VREF - GPIO 1 (3.3 [V])
- AGND and DGND - PIN 39 (GND)

3.5 LED

For the Paper encoded input message, only a single LED of any colour is required (a red LED was used for the project), where as for the LED encoded input message, two different coloured LED's are required (a red and green LED was used for the project).
For this implementation the LED's are connected to a resistor using ports:

$\rightarrow GPIO6$
$\rightarrow GPIO26$
NOTE: any non-specialized GPIO pin could be used.

3.6 Push-Buttons

The push-button is connected using a pull down formation. This was done to account for button debouncing. The button is connected to the raspberry Pi using port:

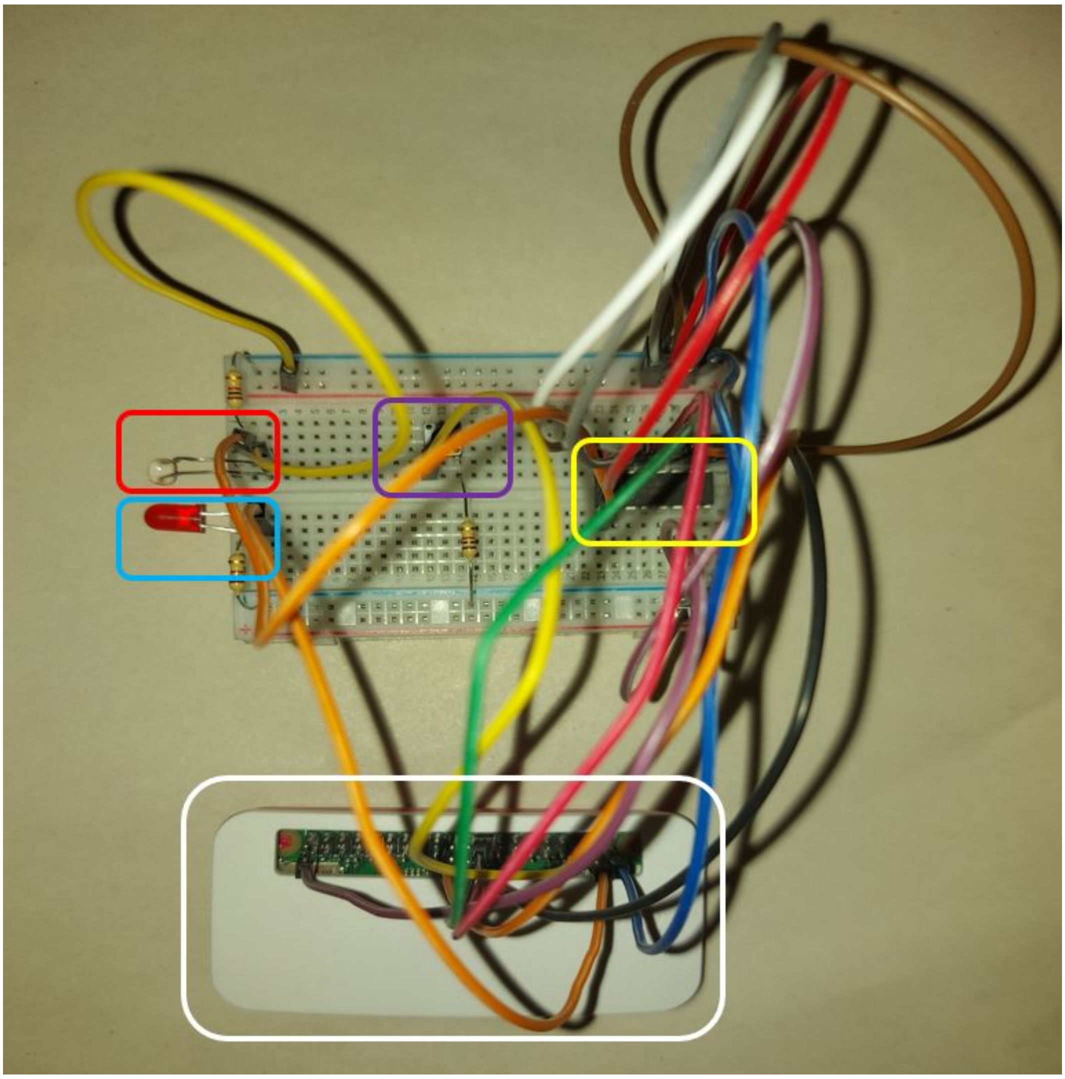
$\rightarrow GPIO16$
NOTE: any non-specialized GPIO pin could be used.

3.7 LDR

The LDR is connected in a voltage divider configuration. This helps to differentiate the difference in voltage levels more accurately resulting in a better reading. The LDR is connected directly to the 3.3 [V] supply at Pin 1 on the Pi. If the levels between the voltages for light and dark light-rays is to small the LDR can be connected to the 5 [V] supply at Pin 2 to give a wider voltage range.

3.8 Initial Setup

Below is the physical set up for the paper encoded message. Note the LED and the LDR are next to each other. For the LED encoded message the two LED's should be adjacent to each other and opposite the LDR, the remainder of the components should be in the same configuration.



The layout of the picture begins with the ADC being on the right-hand-side enclosed by the yellow block. The push-button is in the middle enclosed by the purple block. Finally the LDR enclosed by the red block and the LED enclosed by the blue block are on the left-hand-side of the breadboard. The white block at the bottom is the Raspberry Pi.

References

[1] J.-P. Charras and D. Hollenbeck, "Download." [Online]. Available: https://kicad.org/download/