RPI-3B SPI and Interrupt

David Fransch (FRNDAV011)

Richard Powrie (PWRRIC001)

Practical 4

EEE3096S

2018

1 CONTENTS

I. ANSWERS TO QUESTIONS:

A. EXPLAIN THE SPI COMMUNICATION PROTOCOL WITH A TIMING DIAGRAM

Serial Peripheral Communication (SPI) is used to send data between a microcontroller and peripherals. For this lab the Raspberry Pi used SPI to communicate with an ADC. The SPI protocol is implanted with a master and a slave, where the master controls the clock and decides when to communicate.

In the diagram below is shown a generic timing diagram [1] for SPI communication.

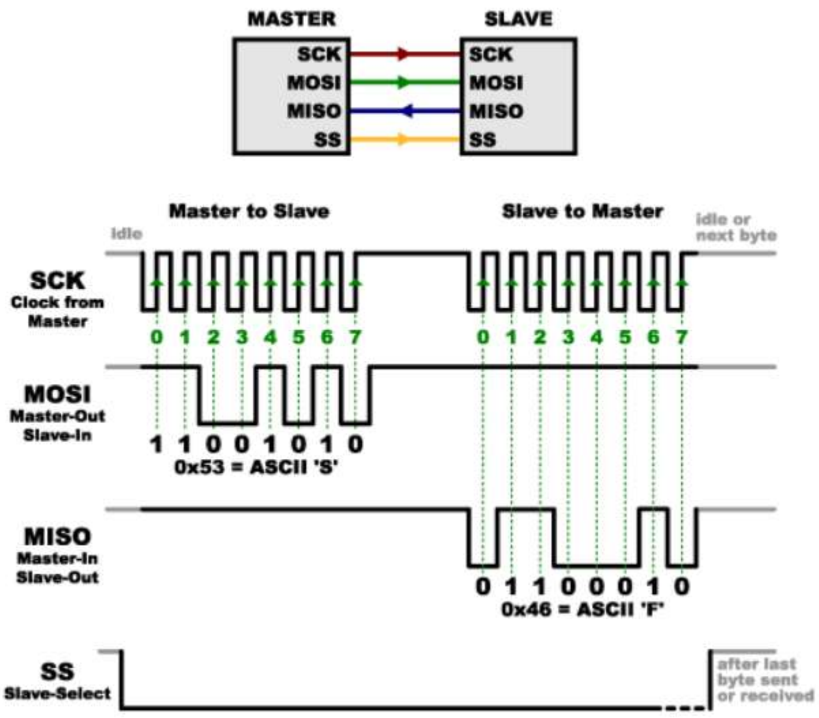
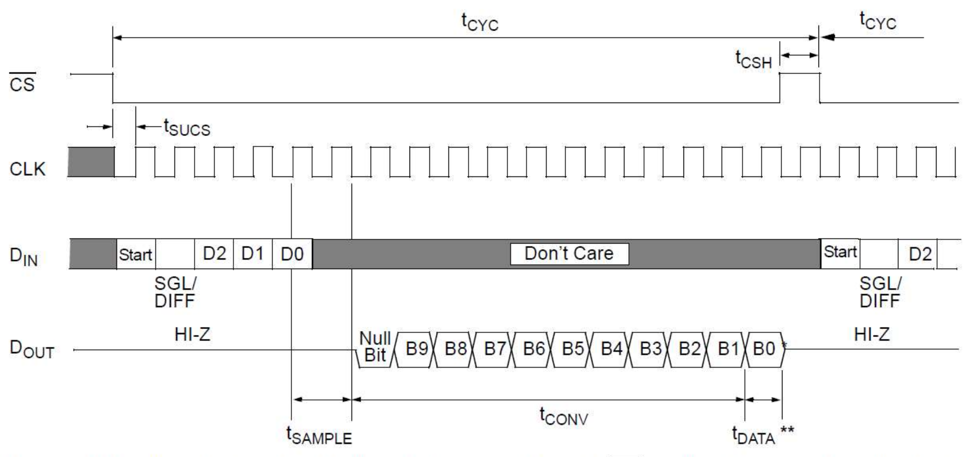


Figure 1: Generic SPI timing diagram

As seen in Figure 1, the master and slave are connected with four lines, SCK, MOSI (Master Out Slave In), MISO (Master In Slave Out), and SS. SCK is the clock sent from the master so that both master and slave know when to sample the data sent to them respectively. MOSI is the serial data channel from master to slave, while MISO is the serial data channel from slave to master. SS is the channel used by the master to select or enable the slave so that it is ready to receive and send data.

Below is the timing diagram taken from the MCP3008 ADC's datasheet [2] for how it receives the commands from the microcontroller.



* After completing the data transfer, if further clocks are applied with $\overline{CS}$ low, the A/D converter will output LSB first data, then followed with zeros indefinitely. See Figure 5-2 below.

** t_{DATA}: during this time, the bias current and the comparator powers down while the reference input becomes a high impedance node.

Figure 2: MCP3008 timing diagram

As seen in Figure 2, for our purposes the master will send 4 bits stating the mode of operation (single or differential inputs) in the first bit and the last three select the channel to be sampled.

B. Define interrupt and threaded call-back in the context of an embedded system.

An interrupt is a process that the microcontroller can have that allows notification of an external IO event happening (e.g. a button being pressed) without the need to constantly poll the IO event to see if it has happened. This allows the main application thread(s) to be running, while the IO thread is running in the background.

A threaded callback function uses one thread to be in charge of IO handling. If an event occurs, the main thread(s) are interrupted (paused) in order to handle the IO event using the related callback function for the specific event. The callback function itself is the function called when the IO thread triggers on an event happening

C. Write a function that converts a 10-bit ADC reading from the potentiometer to a 3V3 limited voltage output.

```
def GetData(channel):#channel = integer 0-7
    adc = spi.xfer2([1,(8+channel)<<4,0])#send three bits, single ended mode
    data = ((adc[1]&3)<<8) + adc[2]
    return data

# function to convert data to voltage level,
# places: number of decimal places needed
def ConvertVolts(data,places):
    volts = (data * 3.3) / float(1023)
    volts = round(volts,places)
    return volts
```

D. WRITE A FUNCTION THAT CONVERTS A 10-BIT ADC READING FROM THE TEMPERATURE SENSOR TO A READING IN DEGREE CELSIUS (HAVE A LOOK AT THE DATASHEET).

```
def convertTemp(data, places):
   temp = (data-0.5)/0.010
   #temp = ((data*330)/float(1023))-50
   temp = round(temp,places)
   return temp
```

E. WRITE A FUNCTION THAT CONVERTS A 10-BIT ADC READING FROM THE LDR TO A PERCENTAGE REPRESENTING THE AMOUNT OF LIGHT RECEIVED BY THE LDR. (2) *THE FLASHLIGHT FROM A SMARTPHONE COULD BE USED AS THE MAXIMUM AMOUNT OF LIGHT RECEIVED BY THE LDR.*

```
def convertLight(data, places):
            #max volt output is 3.11 (from measurement)
            volt = ConvertVolts(data,places)

            return volt/3.11*100
```

F. Draw a flowchart of the system.

Shown below is the flow chart of the system.

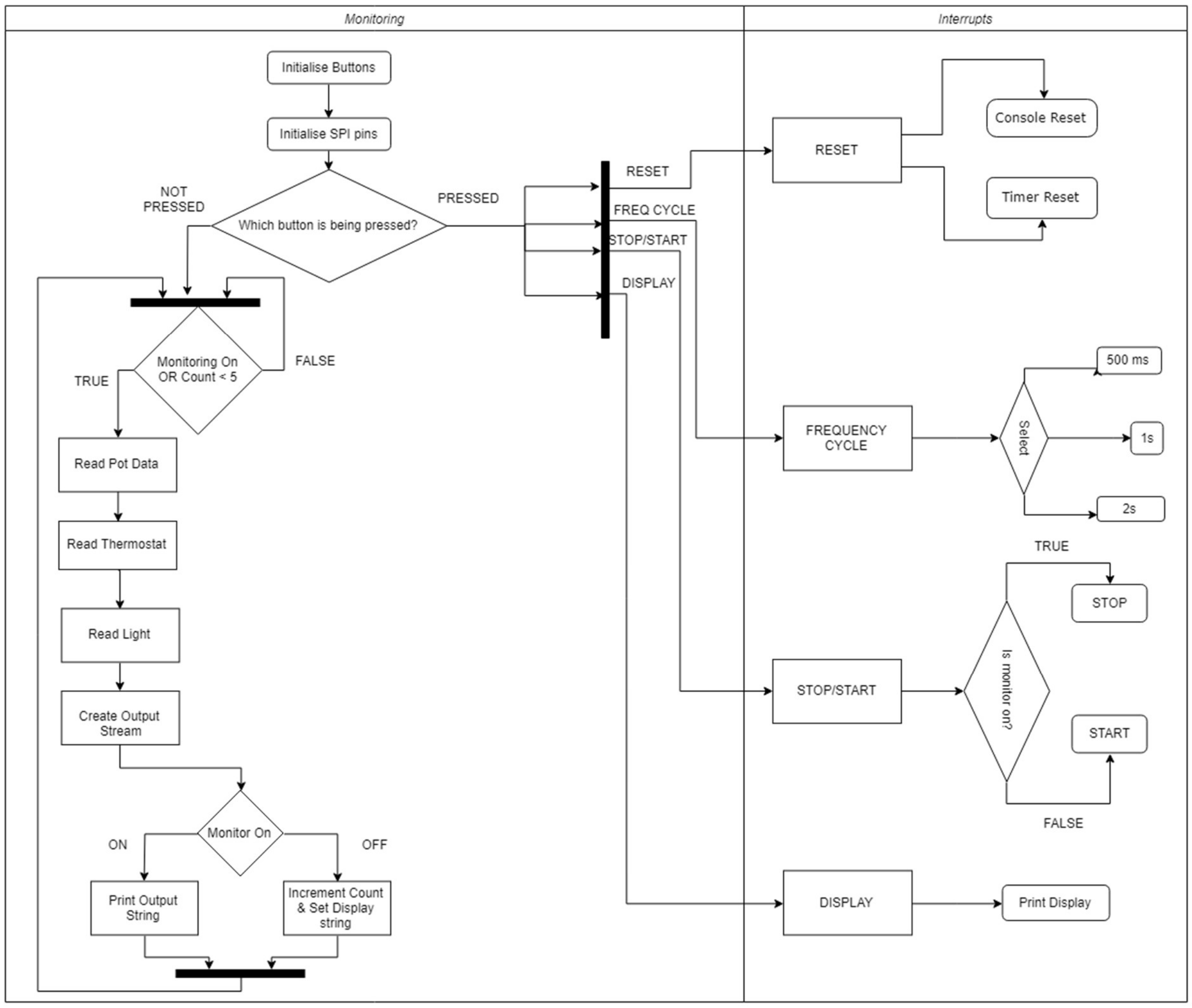


Figure 3: flow chart of system

2 References

[1] Mikegrusin, "Serial Peripheral Interface (SPI)," 1 9 2018. [Online]. Available: https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi.

[2] Microchip, "MCP3004/3008 datasheet," 2007. [Online]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/194715/MICROCHIP/MCP3008.html. [Accessed 1 9 2018].