University of Cape Town

EEE3096S

Embedded Systems

RPi Practical 4

Authors:
Ben Adey
Jonti Oehley

Student Numbers:
ADYBEN002
OHLJON001

September 17, 2018

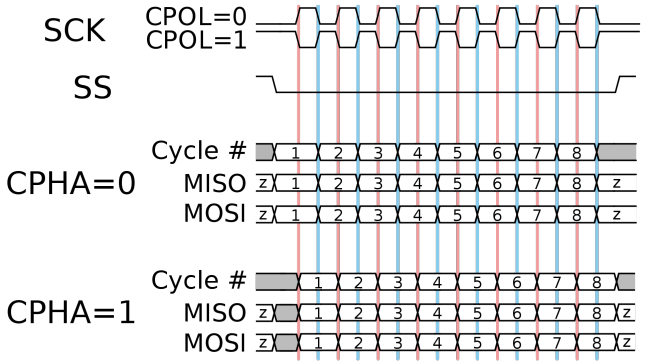
Contents

1 Answers

a SPI Communication, Timing and Modes

Serial Peripheral Interface or SPI is a serial synchronous type of communication through which the two devices in communication share a clock signal to synchronous the transfer of data between them. The device that generates the clock signal shared by the devices is called the master. The master enables the slave device through setting the slave/chip select line low. The SPI hardware on each device contains a shift register. The registers are connected in a loop so that when the master shifts a bit of data to the slave via the Master Out Slave In (MOSI) line, it also receives a dummy piece of data on the Master In Slave Output line. A similar thing is true when the master is receiving, it will then send a dummy piece of data.

The timing of SPI communication can be summarized as four modes which are determined by two settings for the communication. The clock polarity (CPOL) can be either high or low and indicates the state of the clock in its idle state. The clock phase (CPHA) determines whether the input is sampled on the first edge and on every cycle after that or on the second edge and on every cycle afterwards. This is shown graphically in the timing diagram below; The CPOL is high if the clock is



high in idle state and the CPHA is low if the input is sampled on the first edge and high if the input is sampled on the second edge.

b Interrupts and threaded call-backs

An interrupt is a signal to the processor from hardware or software indicating that an event needs immediate processing. The interrupt halts the current operation of the program while an Interrupt Service Routine (ISR) is run. This function contains

the code needed to deal with the event that triggered the interrupt. The ISR runs in supervisor mode.

Threaded callbacks are specific functions that execute under particular external or internal interrupters. The major difference to interrupts is that thread callback functions are executed in a separate thread to the main program. This means that both the callback function and the main function are executing simultaneously in more than one thread.

c Convert 10-bit ADC to 3V3 limited voltage output

```
def conv_10bit_to_3V3(val):
#converts a 10 bit ADC value to a voltage in range 0 - 3.3V
    return round(val*3.3/1023,1)
```

Listing 1: Convert 10-bit ADC value to 3V3 limited voltage output

d Convert 10-bit ADC to reading in degrees Celsius

The MCP9700 datasheet gives the following equation for calculating temperature [1]:

$$V_{out} = T_c \times T_A + V_{0°C}$$

Rearranging to solve for the ambient temperature T_A we get the equation:

$$T_A = \frac{V_{out} - V_{0°C}}{T_c}$$

This is implemented in python code in the following function:

```
def conv_10bit_to_deg_celsius(val):
#converts a 10 bit ADC value to a temperature in degrees Celsius
    return int(round(( (val*3.3/1023) - 0.5 ) / 0.01))
```

Listing 2: Convert 10-bit ADC value to temperature in degrees Celsius

e Convert 10-bit ADC to percentage light value

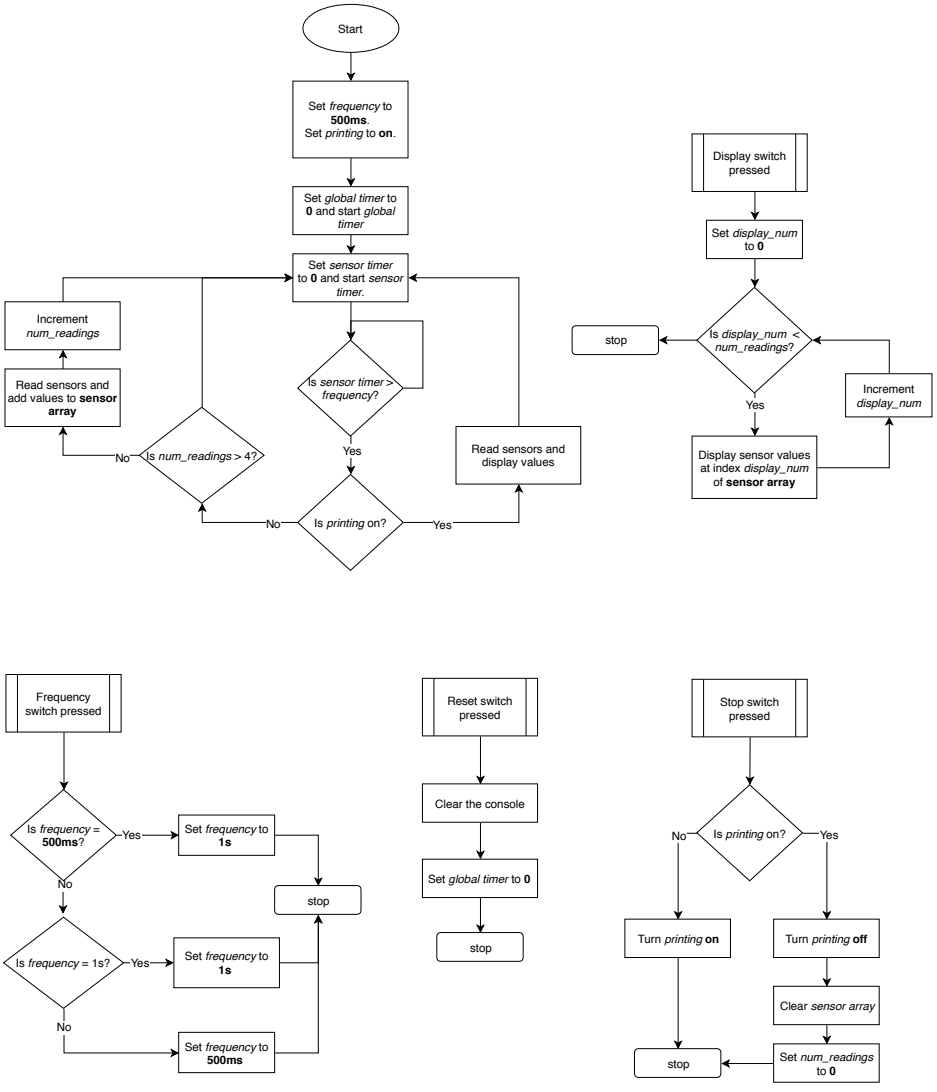
This conversion function uses the variable **max_brightness** which is set to the ADC value when a phone flashlight is shone directly onto the LDR. Thus, this function is expected to return a value of 100 when the phone flashlight is shone on the LDR, and a value of 0 when the LDR is completely covered.

```
max_brightness = 970      #calibrated with phone flashlight

def conv_10bit_to_perc(val):
#converts a 10 bit ADC value to a percentage
    return int(round(val*100/max_brightness))
```

Listing 3: Convert 10-bit ADC value to percentage light on LDR

f Flowchart of the System



Listings

References

[1] Microchip Technology Inc. *MCP9700/9700A and MCP9701/9701A*. Microchip Technology Incorporated, 2016, pp. 1-8.