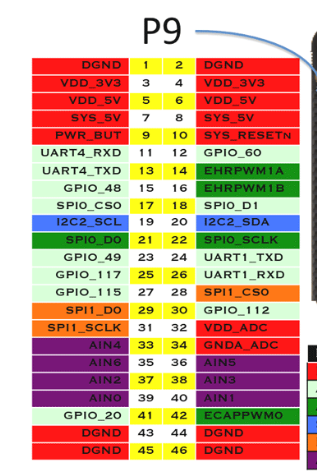
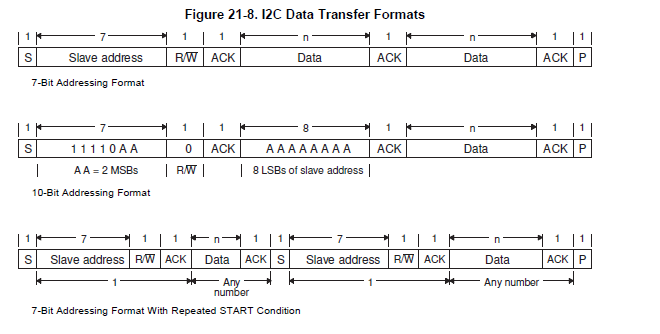
02/23/2020

-Doing some research according to the suggested procedures in the handout and found the following:

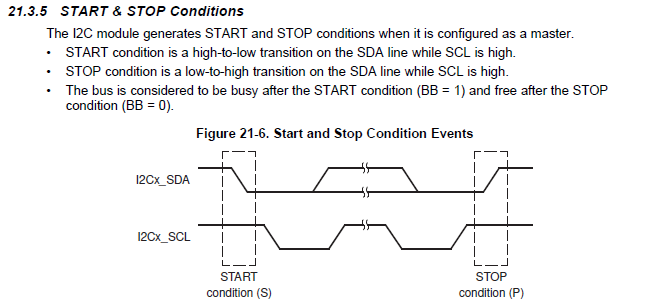


* I2C1\_SCL: 19
* I2C1\_SDA: 20
* conf\_ spi0\_cs0: OFFSET: 95Ch base: 0x44E1\_0000
* conf\_ spi0\_d1: OFFSET: 958h base: 0x44E1\_0000
* To turn on the clock on the I2C1:
  + CM\_PER\_I2C1\_CLKCTRL: write 0x2 at OFFSET 48h base: 0x4802\_A000
* *The high and low states of the data line can only change when the clock signal on the SCL line is LOW*
* *START condition is a high-to-low transition on the SDA line while SCL is high.*
* *STOP condition is a low-to-high transition on the SDA line while SCL is high.*
* *The data is transferred with the most significant bit (MSB) first*
* *7 MSB slave address bits and 1 LSB R/nW bit.*
* *If R/nW is 0, the master writes data into the selected slave; if it is 1, the master reads data out of the slave.*
* **
* I2C base 0x4802A000
* Have 12 MHz : I2C\_PSC **B0h** [7:0]
  + x = SystemClockFreq / 12MHz
  + ICLK = SCLK/(psc + 1)
  + psc = (SCLK/ICLK) -1
* 400 Kbps SCL: SCLL, SCLH **B4h**, **B8h** [7:0]
  + *tLOW = (SCLL + 7) \* ICLK time period*
  + *tHIGH = (SCLH + 5) \* ICLK time period.*
* Configure its own address: I2C\_OA **A8h**
  + *[6:0] when XOA (Expand Own Address, I2C\_CON[7]) is cleared to 0.*
  + *[9:7] bits must be cleared to 000 by application software.*
* Take I2C out of reset: I2C\_CON **A4h**:I2C\_EN = 1 [15]
  + Bit 15 = 1
* Slave address: I2C\_SA **Ach**
  + ??
* Number of bytes to be transferd: I2C\_CNT
  + ???
* bus busy read only: I2C\_IRQSTATUS\_RAW **24h**
  + Bit 12 = 1
* Start bits: I2C\_CON **A4h**
  + Write 0x00008601 (start, transmit, master, enable = 1)
* Transmit data ready: I2C\_IRQSTATUS\_RAW
  + Bit 4 = 1
* Write data: I2C\_DATA
  + 7-0 data to be written
* Need to setup I2C for debugging

02/25/2020

* Started writing High and Low level algorithms
* Done writing the algorithms, but having problems understanding the following:
  + Value of SCLK to be used in dividing the clk
  + Value of tlow to be used to adjusting speed
  + How often should I trigger a stop and start condition? Is it after every byte sent I do a stop condition?

02/27/2020

* Figured out how to get tLow and tHigh from Tyler’s email
* Still do not have SCLK
* Started writing C program
* Does HWREG work also for reading regs?
* Done writing all functions required, except clock freq
* Not sure of the order of function calls, should I write to the buffer first or generate stop condition.
* Do I need to poll XDRY?
* 
* Makes sense to stop after writing

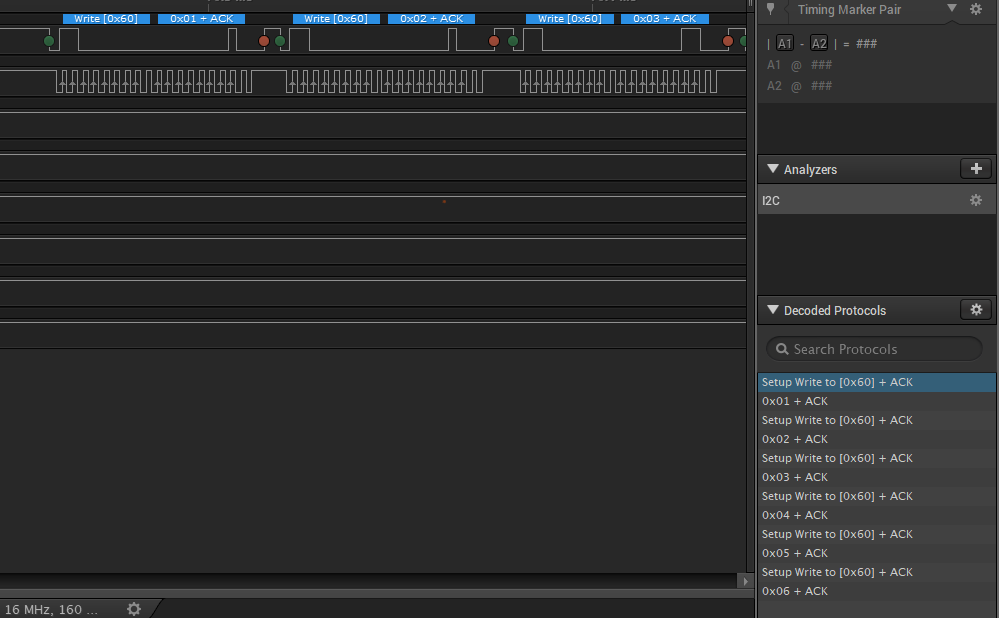
02/26/2020

* Had to turn clk on
* Had to enable stop condition in con reg
* Understood how to set ICLK
* Had to adjust pin mux
* Logic analyzer is 1 indexed

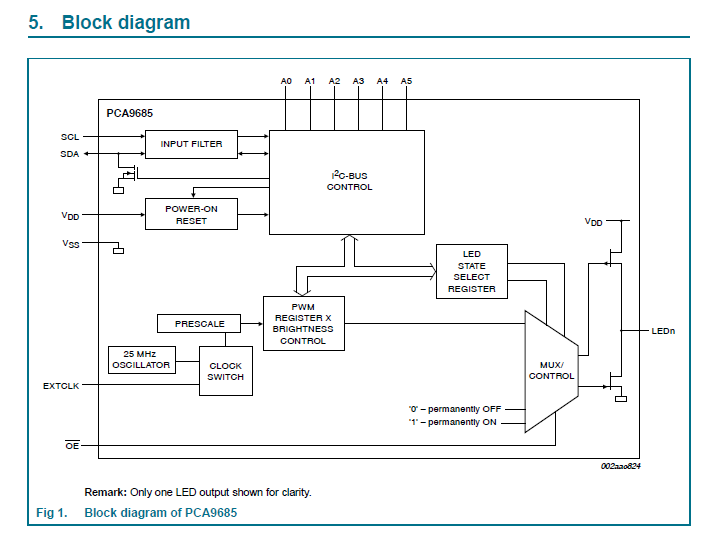
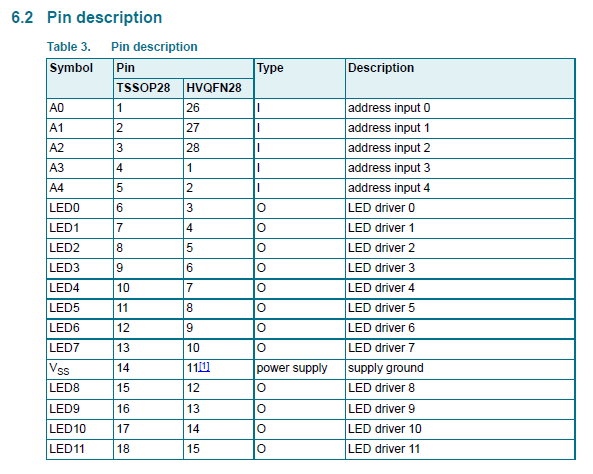
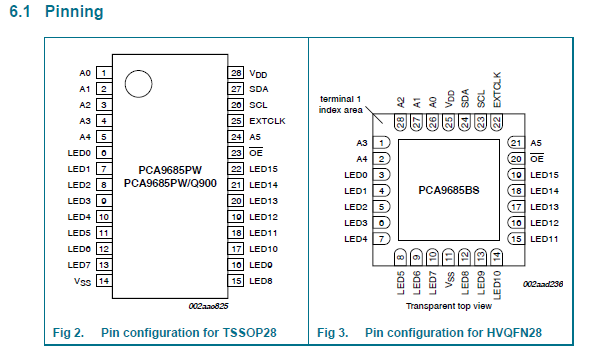
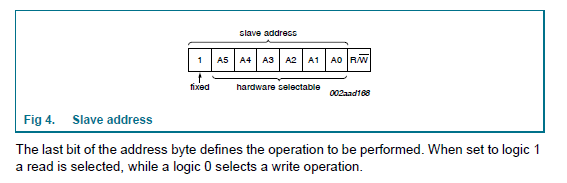
3/6/2020

* Still a problem with the signals produced
* 0xFF + NAK is the result when loop over and write several times

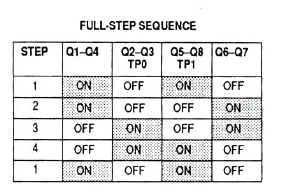
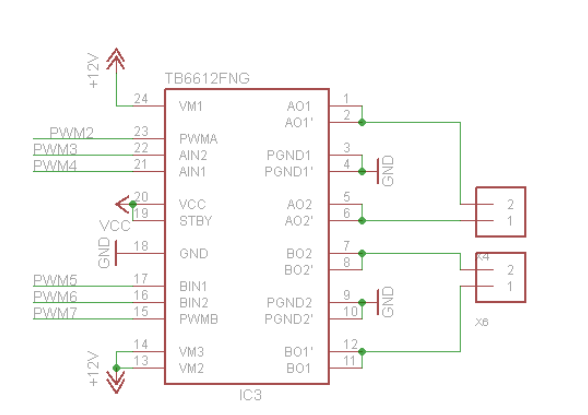
3/7/2020

* Still same problem
* Tried reversing orders not working
* Tried creating a new project not working
* Tried changing stations not working
* Tried putting everything outside the functions and throw in the main not working
* Tired creating an array of data and looping on inside a function WORKING
* PART 1 DONE
* 

3/8/2020

* Started working on Part 2
* PCA sheet findings:
  + The PCA9685 has 4096 steps (12-bit PWM) of individual LED brightness control.
  + The PCA9685 has an external clock input pin that will accept user-supplied clock 50 MHz max.)
  + default state of LEDn output pins is LOW
  + 
  + 
  + 
  + 
  + The turn-on time of each LED driver output and the duty cycle of PWM can be controlled independently using the LEDn\_ON and LEDn\_OFF registers.
  + There will be two 12-bit registers per LED output. These registers will be programmed by the user. Both registers will hold a value from 0 to 4095. One 12-bit register will hold a value for the ON time and the other 12-bit register will hold the value for the OFF time.
  + The PRE\_SCALE register can only be set when the SLEEP bit of MODE1 register is set to

logic 1.

* + Mode1[Sleep]: 00h bit 4 = 1
  + Mode1[ALLCALL]: 00h bit 0
  + Prescale FEh = (25MHz / 4096 \* 1KHz) – 1 = 5 decimal
  + Mode1[restart]: 00h bit 7
  + Mode1[ALLCALL]: 00h bit 0
  + Mode1[EXTCLK]: 00h bit 6 = 0
  + 10x0xxx1
  + Mode2[totem] 01h bit 2 = 1
  + Mode2[INVRT] 01h bit 4 = 0
  + ALL\_LED\_OFF\_H FDh = 0
  + LED\_ON page 21
    - 07h LED0\_ON\_H
    - 0Bh LED1\_ON\_H
    - 0Fh LED2\_ON\_H
    - 13h LED3\_ON\_H
    - 17h LED4\_ON\_H
    - 1Bh LED5\_ON\_H
    - 1Fh LED6\_ON\_H
    - 23h LED7\_ON\_H
  + 
  + 
  + 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | PWM2(PWM) | PWM3(IN2) | PWM4(IN1) | PWM5(IN1) | PWM6(IN2) | PWM7(PWM) |
| A-ON | H | L | H | H | L | L |
| B-ON | L | L | H | H | L | H |

* + Should both be turned on at the same time?
  + H = 0x10
  + L = 0x00