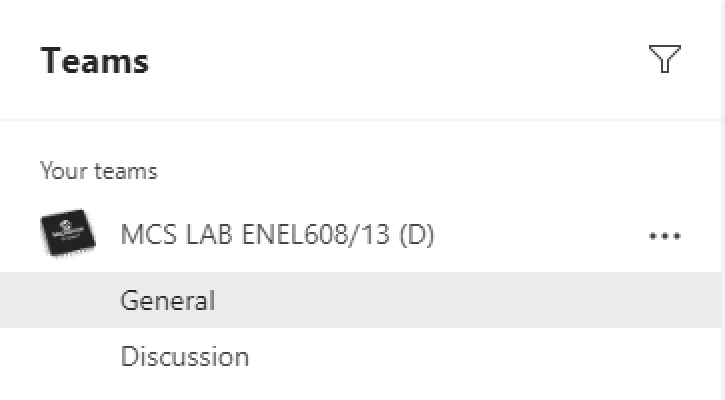
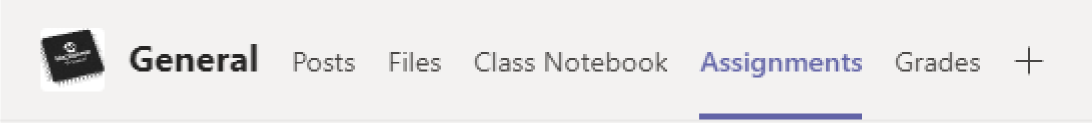
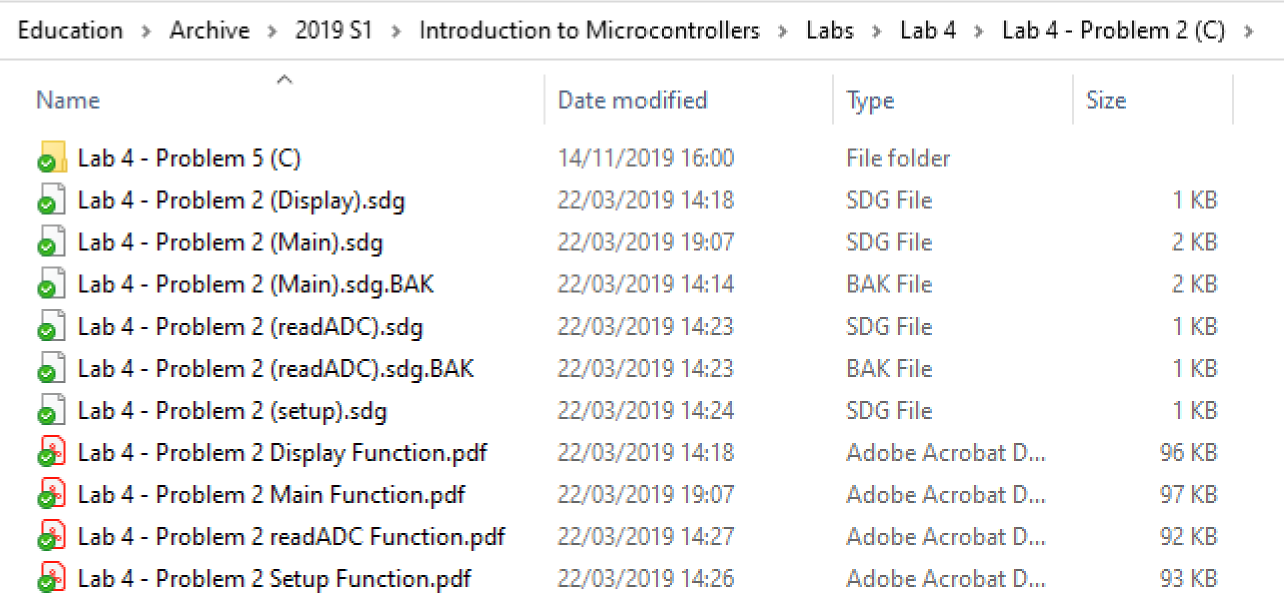
Lab Hand In and Grades





- Navigate to your lab’s Teams group
- Select “Assignments” to view the lab problems and submit your lab report
- Select “Grades” to see all your work marked from Lab 5 onwards

Creating Structure Diagrams

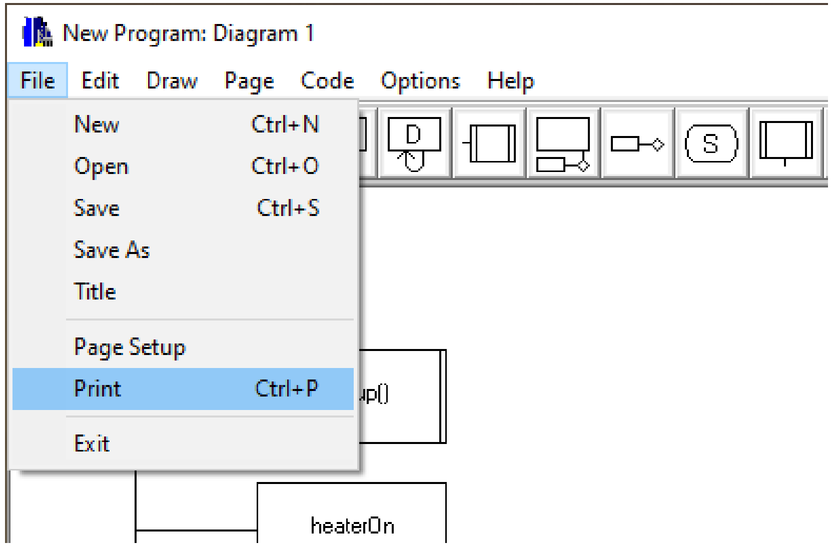


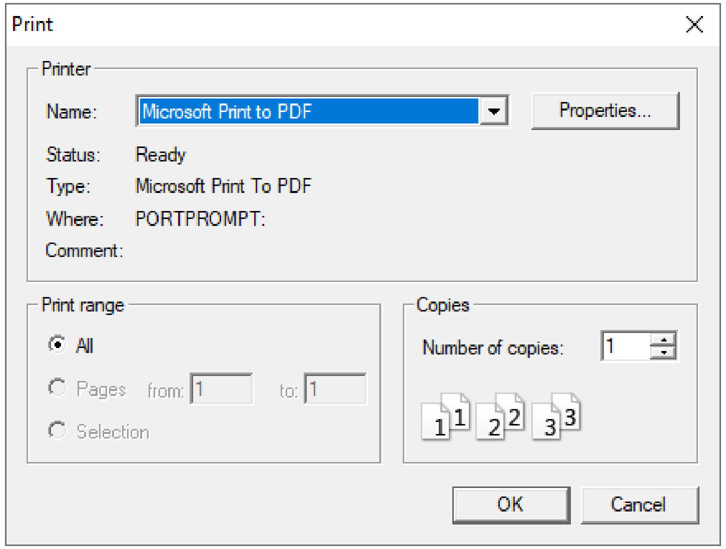
Education › Archive › 2019 S1 › Introduction to Microcontrollers › Labs › Lab 4 › Lab 4 - Problem 2 (C) ›

Name	Date modified	Type	Size
Lab 4 - Problem 5 (C)	14/11/2019 16:00	File folder	
Lab 4 - Problem 2 (Display).sdg	22/03/2019 14:18	SDG File	1 KB
Lab 4 - Problem 2 (Main).sdg	22/03/2019 19:07	SDG File	2 KB
Lab 4 - Problem 2 (Main).sdg.BAK	22/03/2019 14:14	BAK File	2 KB
Lab 4 - Problem 2 (readADC).sdg	22/03/2019 14:23	SDG File	1 KB
Lab 4 - Problem 2 (readADC).sdg.BAK	22/03/2019 14:23	BAK File	1 KB
Lab 4 - Problem 2 (setup).sdg	22/03/2019 14:24	SDG File	1 KB
Lab 4 - Problem 2 Display Function.pdf	22/03/2019 14:18	Adobe Acrobat D...	96 KB
Lab 4 - Problem 2 Main Function.pdf	22/03/2019 19:07	Adobe Acrobat D...	97 KB
Lab 4 - Problem 2 readADC Function.pdf	22/03/2019 14:27	Adobe Acrobat D...	92 KB
Lab 4 - Problem 2 Setup Function.pdf	22/03/2019 14:26	Adobe Acrobat D...	93 KB

- Make use of multiple files, by spreading out functions
- One Function per file will make it easier to spread out content
- You can copy + paste these files too if another problem uses the same function (readADC)

Exporting Structure Diagrams




Print

Printer

Name: Microsoft Print to PDF

Status: Ready

Type: Microsoft Print To PDF

Where: PORTPROMPT:

Comment:

Print range

● All

○ Pages from: 1 to: 1

○ Selection

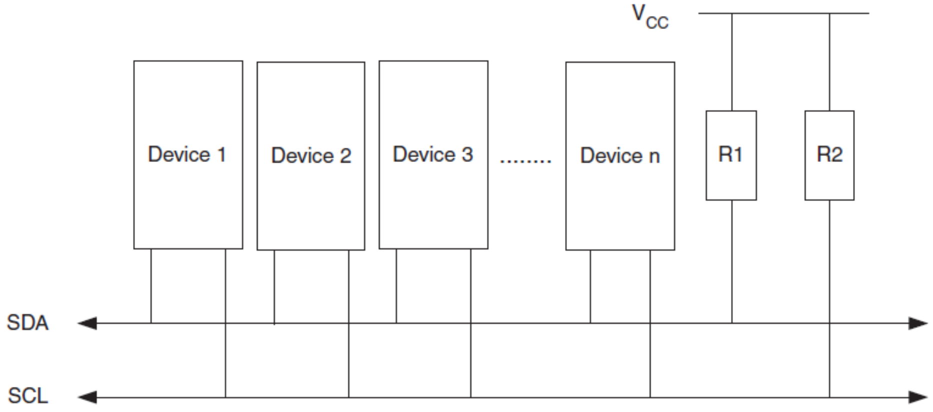
Copies

Number of copies: 1

- Go to “File” -> “Print”
- Select “Microsoft Print to PDF” and Save output file
- Submit PDF with report OR Take screenshot of PDF and attach to report

Two Wire Interface

Figure 21-1. TWI bus interconnection.

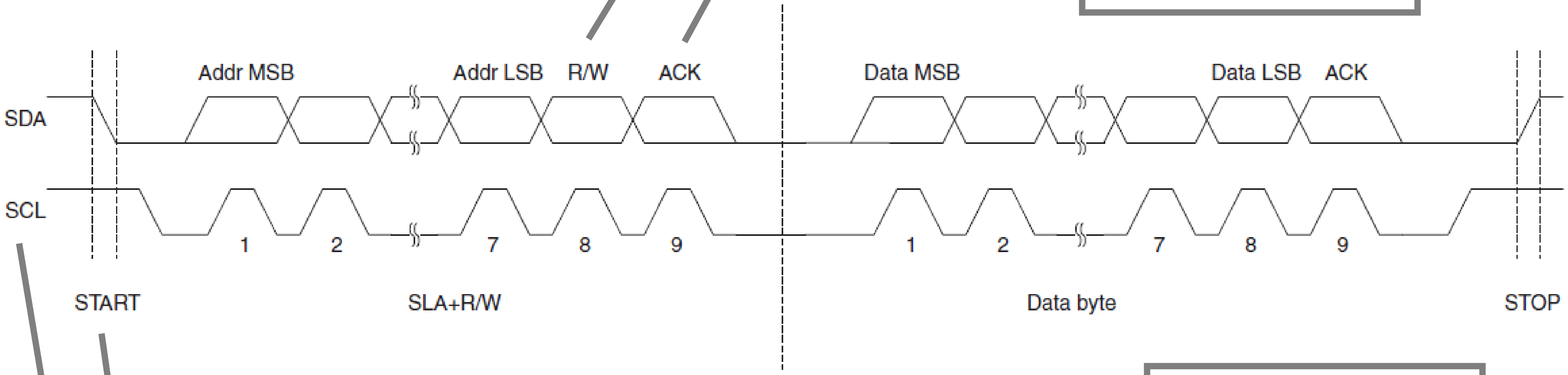


- Half Duplex – Transmit and receive data one at a time
- Synchronous – Shared clock between the Master and Slave
- Mutli-slave Multi-Master
- The only protocol that transmits “Acknowledge” bits
- There is one register that need to be set in the `setup()` to use TWI
  - `TWBR`

Microchip AT90USB1287 Datasheet

TWI – Data Transmission

Figure 21-6. Typical data transmission.



Tell slave if we are to perform read or write

Note: ACK is sent from the receiver

The Start/Stop do not have a clock ‘pulse’

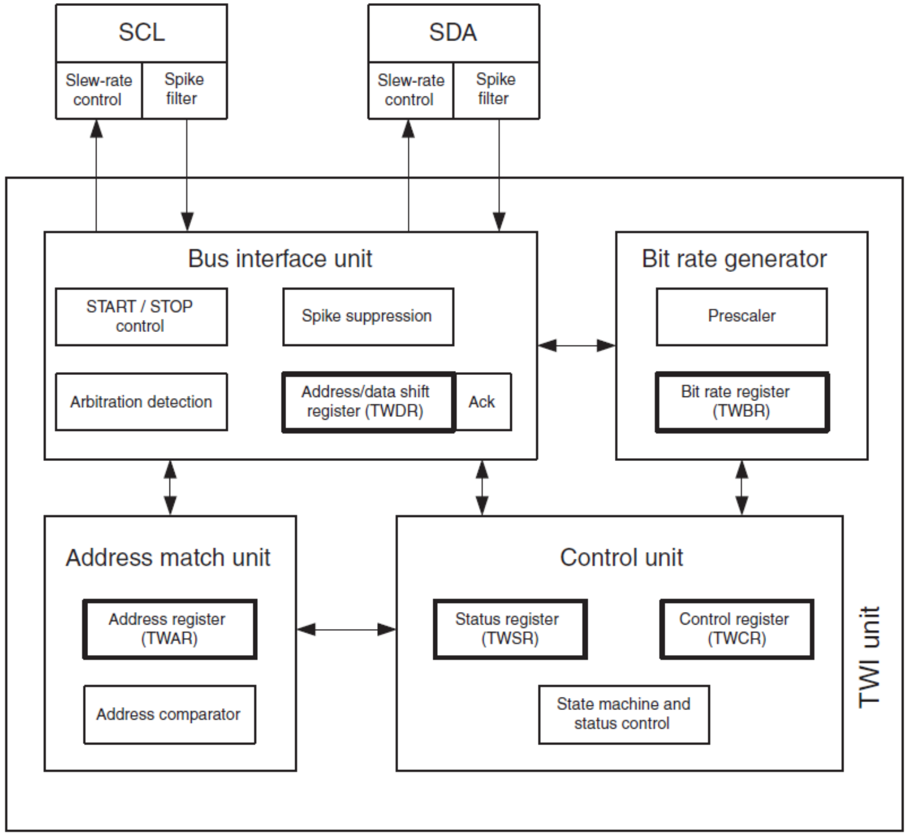
Clock Signal generated by µC, Tx over *SCL* line

Microchip AT90USB1287 Datasheet

SPI – Block Diagram

- The Clock (Bit rate generator)
  - Based off the f_{osc}, and processed through prescaler
  - Set using the TWBR Register
- Bus Interface Unit
  - Controls the flow of data out to the bus
  - Also co-ordinates the clock and data transmission
  - Stores the Address/Data in shift registers
  - Stores the Acknowledge bit

Figure 21-9. Overview of the TWI module.



Microchip AT90USB1287 Datasheet

TWI – Transmission

- Too much to fit on a PowerPoint Slide…

TWITx Function

```
// Function Contrstuctor to Transmit data to a device over TWI with
address 'slaveAddress’, register 'registerAddress’, and payload 'data’

char TWITx(char slaveAddress, char registerAddress, char data);
```

- The TWITx Function completes the process of sending data over TWI
  - It will facilitate the communication to a slave at `slaveaddress`, writing `data` to the register `registerAddress`
- This function returns a char value
  - Take a look at the function and how it works – if the value returned is 0, no issues are encountered in the transmission. If the value returned is 1, an issue has occurred in the TWI transmission and it should be repeated
  - Hint: Use a while loop to run the code until the value returned is a 0!
- The code is provided to you in your Microcontroller Manual (a.k.a `TWIwrite()`)
  - You are not expected to re-create the procedure for sending TWI Data – you can freely copy this function
  - You are expected to make a Structure Diagram of this function
  - You will still need to setup everything else (ex. addresses, format of the data, registers, `#define` statements)
  - A courtesy copy of this function has been provided to you in your Assignment Materials - I have also made detailed comments to help you understand how this works

RGB LED

- This week will require using the I^{2}C LED
  - The I^{2}C LED can be contacted on slave address 0xCC
  - All info is in Appendix H on Pg. 67 of the Lab Manual
- The three colour channels (R,G,B) are controlled individually
  - Each colour allows for 64 steps (~262k potential colours)
  - Each colour corresponds to a group A, B, or C
    - RegA is Green
    - RegB is Red
    - RegC is Blue
  - The chip is capable of driving two sets of LED’s
    - Channel 1 & 2 in our case have been driven together to supply more current to the LED
- Ensure that you enable the LED’s colour groups in your `setup()` by setting RegEn
  - The RegEn register can be contacted on register address 0x00

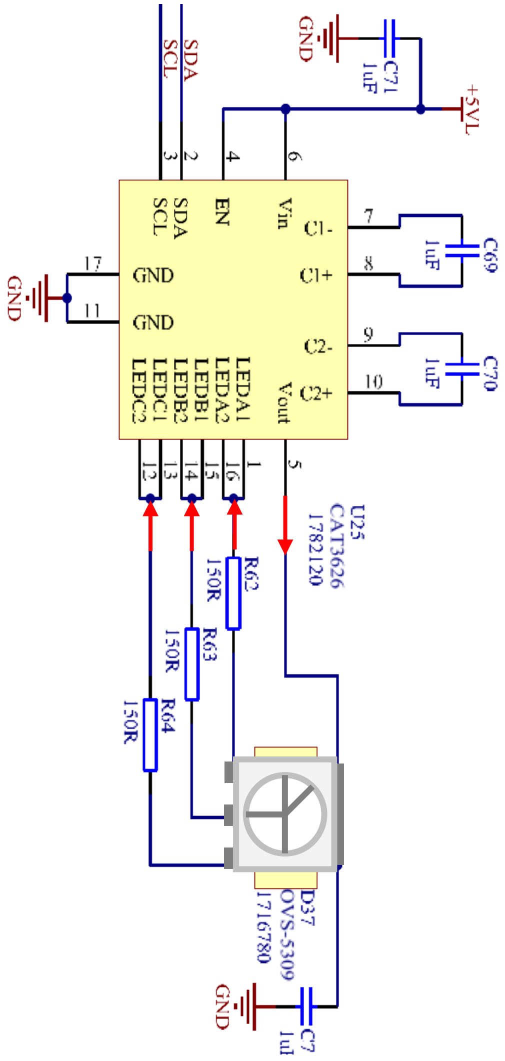


Image Attribution: Mark Beckerleg

RGB LED

- The controller pulls pins down
  - V_{out} is sent from the chip to LED's.
  - Each LEDxx Pin sinks the voltage to ground
  - Capacitors reduce the ripples in the current due to PWM
- One very important note
  - The I^{2}C LED also has functions that automatically configure the TWI and controls the LED channels and colours
  - These commands are part of the Lab board library
  - In this lab, you must manually perform commands to control the LED. **<u>You cannot use any of the library functions to control the RGB LED!</u>**

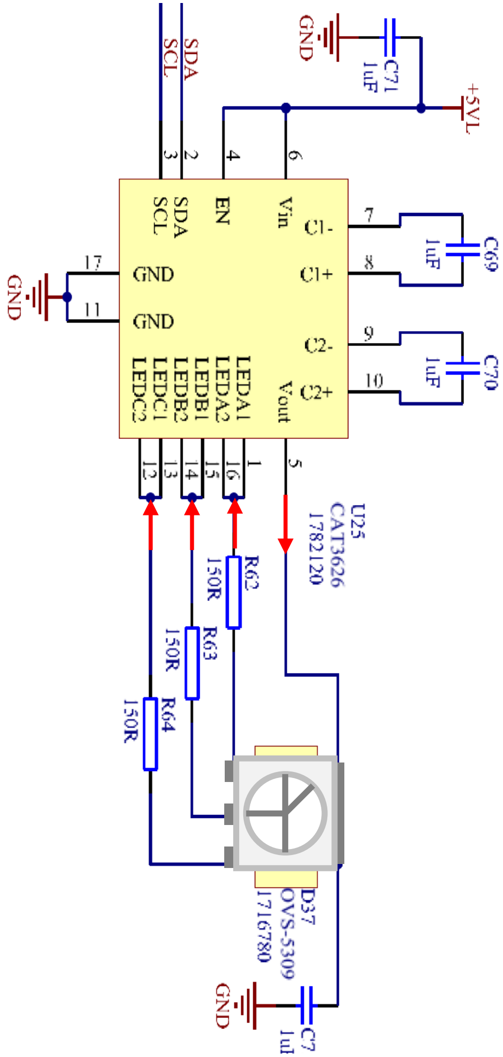
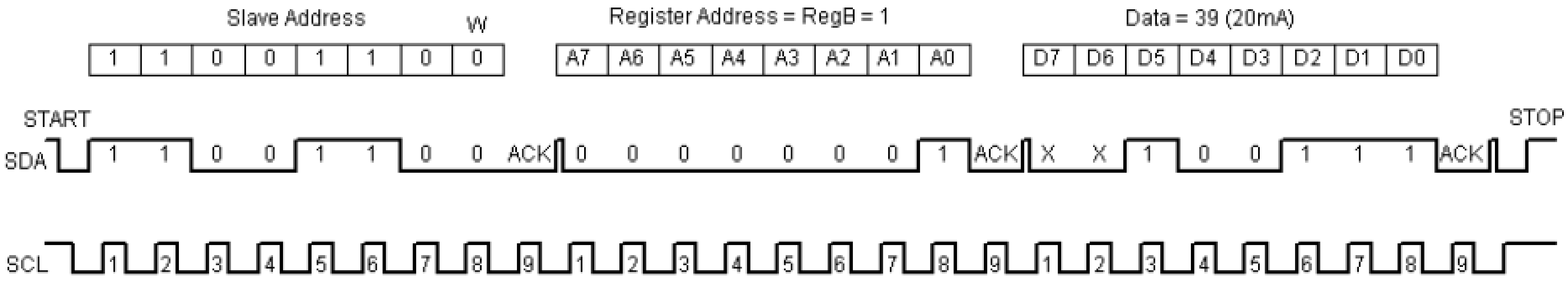


Image Attribution: Mark Beckerleg

RGB LED

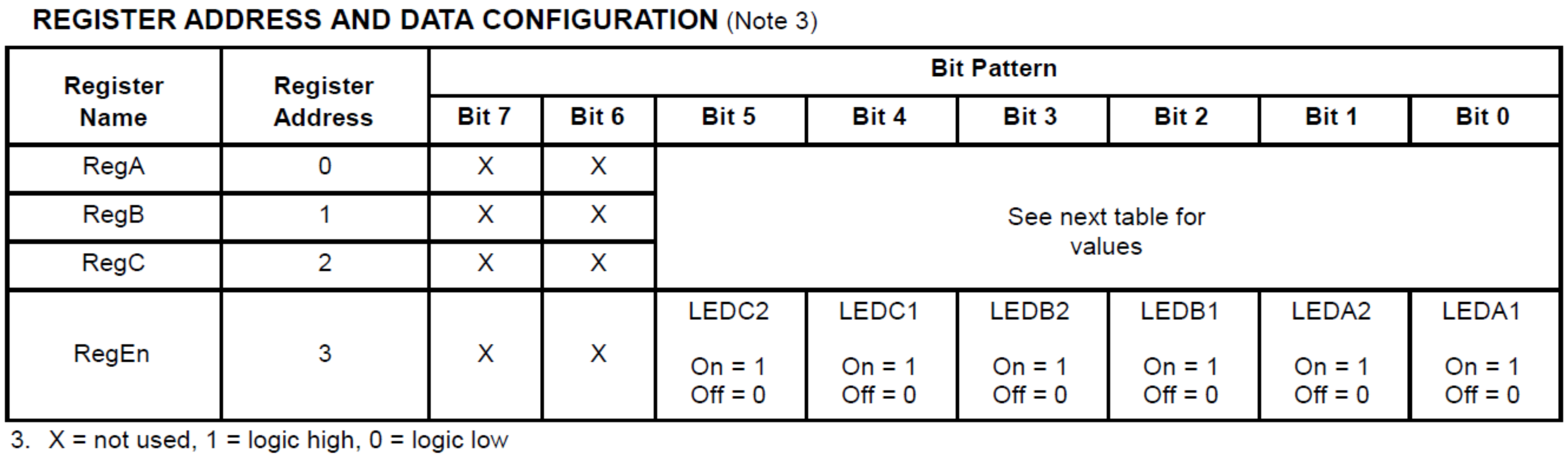


Write Instruction Example Setting 20 mA Current in LEDB1 and LEDB2

- An example of transmitting data over the TWI
  - Note: The TWITx Function completes the process of sending data over TWI (you don’t need to stress over this!)
- Three blocks of data sent
  - Slave Address
  - Register Address
  - Data

Image Attribution: Mark Beckerleg

RGB LED



REGISTER ADDRESS AND DATA CONFIGURATION (Note 3)

Register Name	Register Address	Bit Pattern							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RegA	0	X	X	See next table for values					
RegB	1	X	X						
RegC	2	X	X						
RegEn	3	X	X	LEDC2 On = 1 Off = 0	LEDC1 On = 1 Off = 0	LEDB2 On = 1 Off = 0	LEDB1 On = 1 Off = 0	LEDA2 On = 1 Off = 0	LEDA1 On = 1 Off = 0

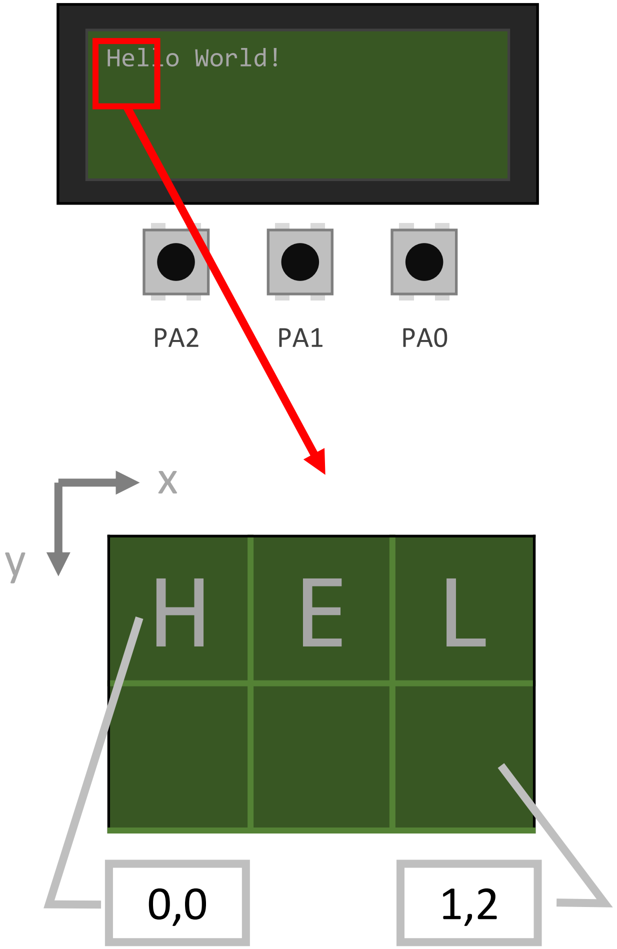
3. X = not used, 1 = logic high, 0 = logic low

- The LED’s have 6-bits of resolution
  - Note: The two Most Significant Bit’s are ignored
- Four Registers
  - Reg A – C: Inidividual Colours
  - RegEn: Enable or disable LED colours

Image Attribution: Mark Beckerleg

LCD Display

- This week will require using the 20x4 Alphanumeric display
  - Review the contents of your lab manual to see how this is setup
  - The example on Page 24 is a good example to refer to
- Call these functions in your `setup()`
  - `SLCDInit()` – This will initiate the TWI serial protocol to allow your µC to talk to the LCD
  - `SLCDDisplayOn()` – This will turn on the display to show characters
  - `SLCDClearScreen()` – This will clear the screen of any existing content (such as the last students code)
- The LCD is split into a 20 x 4 grid
  - The top-left cell is position 0,0
  - Use the `SLCDSetCursorPosition(y,x)` function to set the cursor position
  - Use the `SLCDWriteString(charArrayToWrite)` function to place text on the screen
  - The text needs to be in the form of a `char` array (use the `sprintf()` function)



sprintf Function

```
// Example of using the sprintf function

char line0[20];
sprintf(line0, "temperature - %3d",tempSensorValue);
```

- ‘`line0`’ is the char array in which the string will be stored in
  - This should be defined prior (as per the example above), and we set this to a blank array of 20 char values (since each line of the display can only show 20 characters)
- ‘`temperature - %3d`’ is the string we want to convert to show on the display
  - %3d indicates that we intend to reserve three (‘3’) spaces for a decimal number (‘d’) to be displayed
- ‘`tempSensorValue`’ is the number we plan to substitute in this location
  - The space we reserved earlier will be replaced by a 3-digit decimal number
- Multiple variables can be used for the sprintf function
  - See your lab manual for more information