Inter-Integrated Circuit (I2C)

Lab 6

Readings:

- Lecture 15 - I2C
- I2C_DAC_MAX517-MAX519 datasheet
- I2C_display datasheet

Lab Description:

In this Lab students will be required to communicate to the I2C DAC and the I2C LCD module. This will be done through the SAMD20's I2C **interrupt method not the I2C polling method. PA08 and PA09** will be used to control the SDA and SCL bus lines respectively. PA08 and PA09 will need to be set for SERCOM0 or peripheral C.

Some registers need synchronization when writing to them. The following list has the registers and bits that need synchronization.

- Data (DATA) when in smart mode
- Software Reset bit in the Control A register (CTRLA.SWRST)
- Enable bit in the Control A register (CTRLA.ENABLE)
- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in master operation

MAX518 (I2C DAC): (LED)

The MAX518 is an I2C DAC with two outputs. It requires an ADDRESS byte followed by a COMMAND byte and then the data OUTPUT byte. The figure below shows the complete serial Transmission.

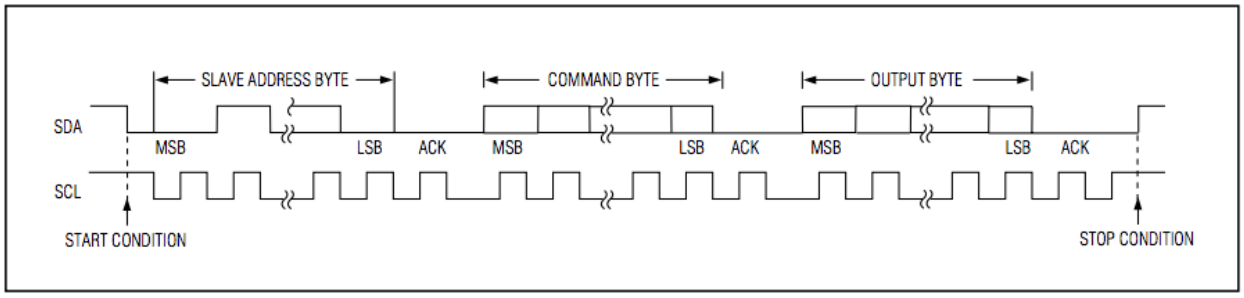

Figure 4. A Complete Serial Transmission

The slave address consists of a 7 bit address followed by 1 bit to determine if the following bytes are to be written or read.

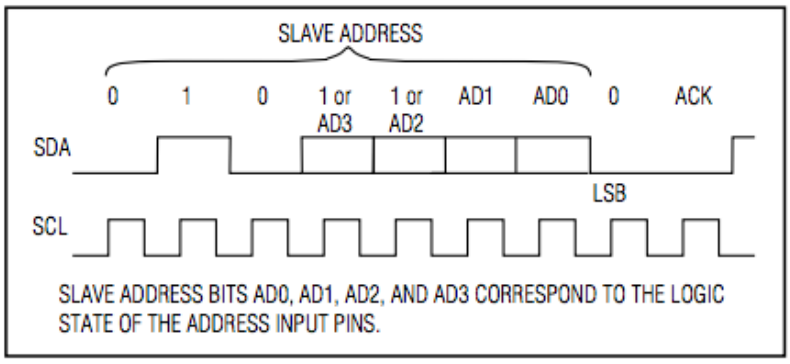

Figure 6. Address Byte

The Command byte is used to determine which output of the DAC is to be used. The last bit of the command byte determines OUT0 or OUT1 by having a 0 or 1 respectively.

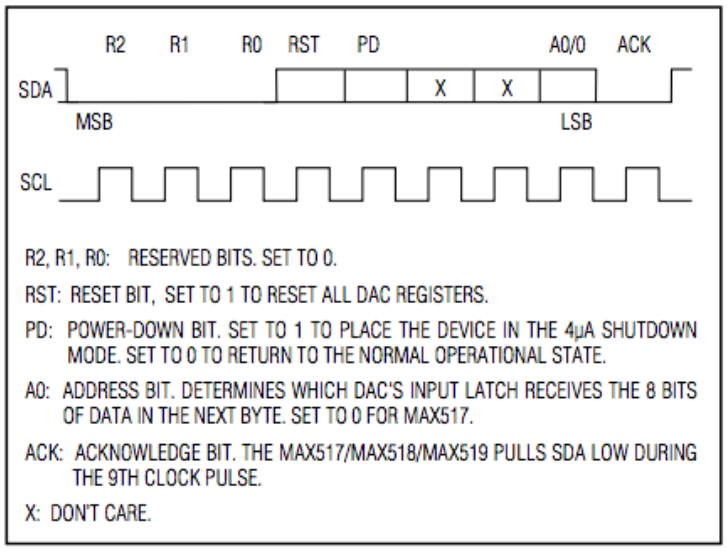

Figure 7. Command Byte

I2C LCD display:

The I2C LCD display needs the following for proper communication. A slave address with a read or write bit followed by a control byte followed by data. The control has a rs bit and a co bit. The co bit determines if another control byte is to be sent after the first data byte. A 0 means a control byte will not follow the data byte. A 1 means a control byte will be following the next data byte. The rs bit determines if the DRAM is to be accessed. A 0 means other instructions will be accessed instead of the DRAM. A 1 means the data byte will be written to the DRAM.

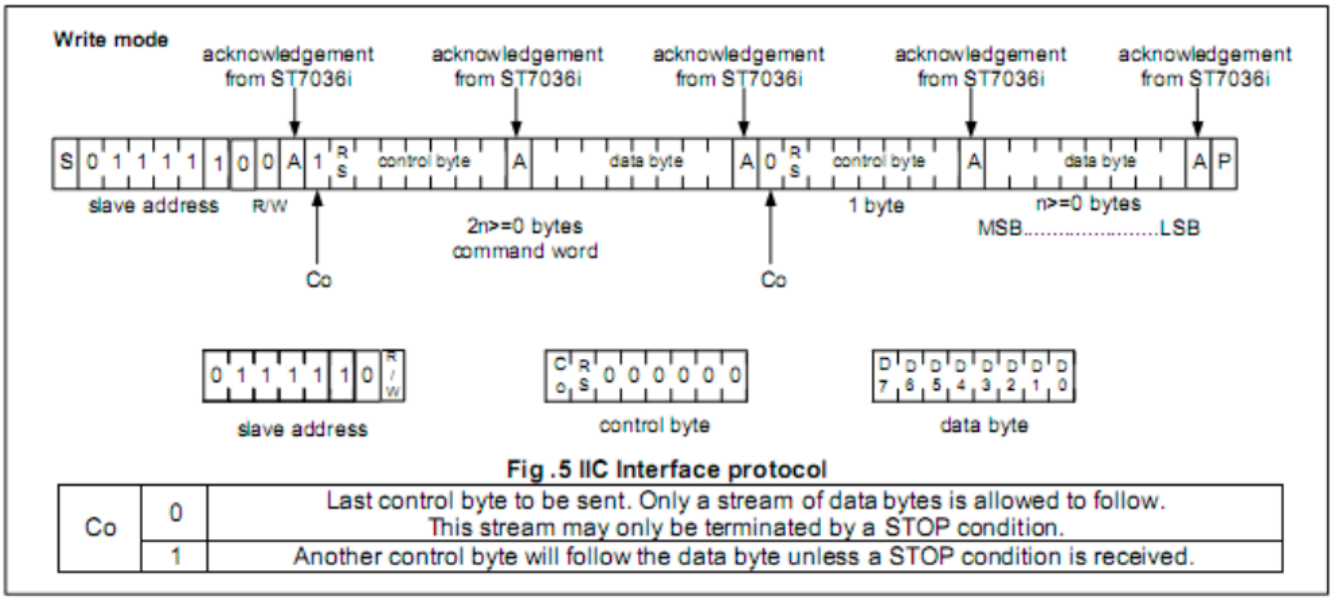

Fig .5 IIC Interface protocol

Co	0	Last control byte to be sent. Only a stream of data bytes is allowed to follow. This stream may only be terminated by a STOP condition.
	1	Another control byte will follow the data byte unless a STOP condition is received.

When the number of display characters is less than 40 x 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the ST7036 is used, 20 characters x 2 lines are displayed. See Figure 9. When display shift operation is performed, the DDRAM address shifts.

Display Position

	1	2	3	4	5	6		38	39	40
DDRAM Address (hexadecimal)	00	01	02	03	04	05	.....	25	26	27
	40	41	42	43	44	45	.....	65	66	67

Fig. 9 2-Line Display

Display Position	1	2	3	4	5	6	7	8		17	18	19	20
DDRAM Address	00	01	02	03	04	05	06	07	.....	10	11	12	13
	40	41	42	43	44	45	46	47	.....	50	51	52	53
For Shift Left	01	02	03	04	05	06	07	08	.....	11	12	13	14
	41	42	43	44	45	46	47	48	.....	51	52	53	54
For Shift Right	27	00	01	02	03	04	05	06	.....	0F	10	11	12
	67	40	41	42	43	44	45	46	.....	4F	50	51	52

Fig. 10 2-Line by 20-Character Display Example

The following Table of Commands list all of the instructions that can be used for the LCD Display. The pseudo code at the end of this lab manual shows how to initialize the LCD display. You can use that code with this table to understand what the instructions do.

Table of Commands

Instruction	Instruction Code										Description
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC
Return Home	0	0	0	0	0	0	0	0	1	x	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	D=1:entire display on C=1:cursor on B=1:cursor position on
Function Set	0	0	0	0	1	DL	N	DH	IS2	IS1	DL: interface data is 8/4 bits N: number of line is 2/1 DH: double height font IS[2:1]: instruction table select
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM/ICONRAM)
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM/ICONRAM)

Instruction table 0(IS[2:1]=[0,0])											
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	X	X	S/C and R/L: Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.
Set CGRAM	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter

Instruction table 1(IS[2:1]=[0,1])											
Bias Set	0	0	0	0	0	1	BS	1	0	FX	BS=1:1/4 bias BS=0:1/5 bias FX: fixed on high in 3-line application and fixed on low in other applications.
Set ICON Address	0	0	0	1	0	0	AC3	AC2	AC1	AC0	Set ICON address in address counter.
Power/ICON Control/ Contrast Set	0	0	0	1	0	1	Ion	Bon	C5	C4	Ion: ICON display on/off Bon: set booster circuit on/off C5,C4: Contrast set for internal follower mode.
Follower Control	0	0	0	1	1	0	Fon	Rab 2	Rab 1	Rab 0	Fon: set follower circuit on/off Rab2~0: select follower amplified ratio.
Contrast Set	0	0	0	1	1	1	C3	C2	C1	C0	Contrast set for internal follower mode.

Instruction table 2(IS[2:1]=[1,0])											
Double Height Position Select	0	0	0	0	0	1	UD	X	x	x	UD: Double height position select
Reserved	0	0	0	1	X	X	X	X	X	X	Do not use (reserved for test)

Use the following table to determine what to write to the DRAM to display a character to the current cursor location. *<u>hint:</u> All of the numbers and letters follow the ASCII code which is the same as a char of that letter or number. example if an uppercase 'A' needed to be displayed the following would need to be written to the DRAM 0b01000001, or 0x41, or 'A' (a char)*


b7-b4
b3-b0
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

Peripheral and Coding:

Address-		SAMD20 Syntax Code	
0x42000800	-	SERCOM0	// address for SERCOM0
offset 0x00	-	CTRLA.reg	// Set MODE, SDAHOLD, and PINOUT
offset 0x04	-	CTRLB.reg	// use to enable smart mode and send stop bit with CMD
offset 0x0A	-	BAUD.reg	// use to set the baud rate
offset 0x0D	-	INTENSET.reg	// use to enable interrupts
offset 0x10	-	STATUS.reg	// used for synchronization
offset 0x14	-	ADDR.reg	// use to write address
offset 0x18	-	DATA.reg	// use to send or read data

Required Tasks: ***please comment your code**

For all tasks implement the I2C using **interrupts** not by polling

Task 1: Using the MAX518 I2C DAC IC turn on and off the LED from the IC's output 0 (OUT0)

- the 7 bit slave address of the I2C DAC is 0x2C or 0b010 1100
- screenshot the I2C communication using the oscilloscope (include address, command, and data)

EC: Create your own I2C driver hint: it should look the same as your waveform from Task 1.

Task 2: Communicate to the LCD module and have it display your group members names.

- separate the names by having one on the top line and the other name on the bottom line
- PB30 is connected to the reset pin of the LCD display which is active low, so set PB30 to high
- the 7 bit address of the LCD module is 0x3C or 0b011 1100

hint: check the pseudo-code listed below to start up the LCD screen
hint2: create a function that takes in a string to output to the I2C bus one char at a time

```
/*****************************************************
*           Initialization For LCD Display          *
*****************************************************/
void init_LCD(){
I2C_Start();// Start bit
I2C_out(Address); // Address
I2C_out(Comsend);//send a command of 0x00 so that command does not have to be
                   sent again
I2C_out(0x38);
delay(10);
I2C_out(0x39);// Function Set
delay(10);
I2C_out(0x14);// Bias Set
I2C_out(0x78);// Contrast Set
I2C_out(0x5E);// Power/Icon Control
I2C_out(0x6D);// Follower Control
I2C_out(0x0C);// Display ON/OFF (data bit 1 will turn the cursor on or off)
I2C_out(0x01);// Clear Display
I2C_out(0x06);// Entry Mode
delay(10);
I2C_Stop();// Stop bit
}
/*****************************************************/
```