Implementing a:
Gesture Activated Camera
using the ATMEL SAM V71 Xplained

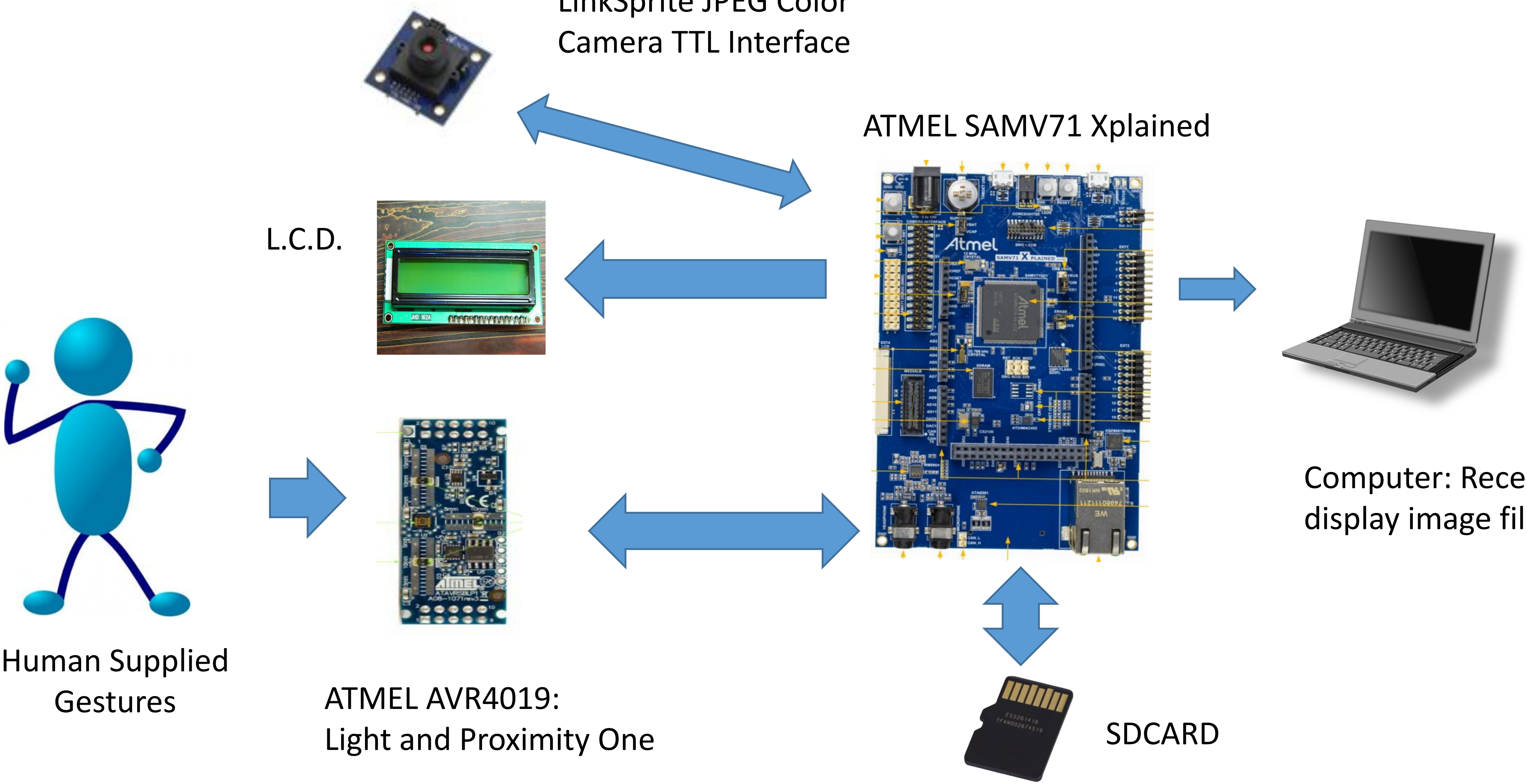
Tom DesRosiers

Lawrence Technological University

EEE 6994: System on a Chip

Spring 2016

Hardware Overview:



LinkSprite JPEG Color Camera TTL Interface

ATMEL SAMV71 Xplained

L.C.D.

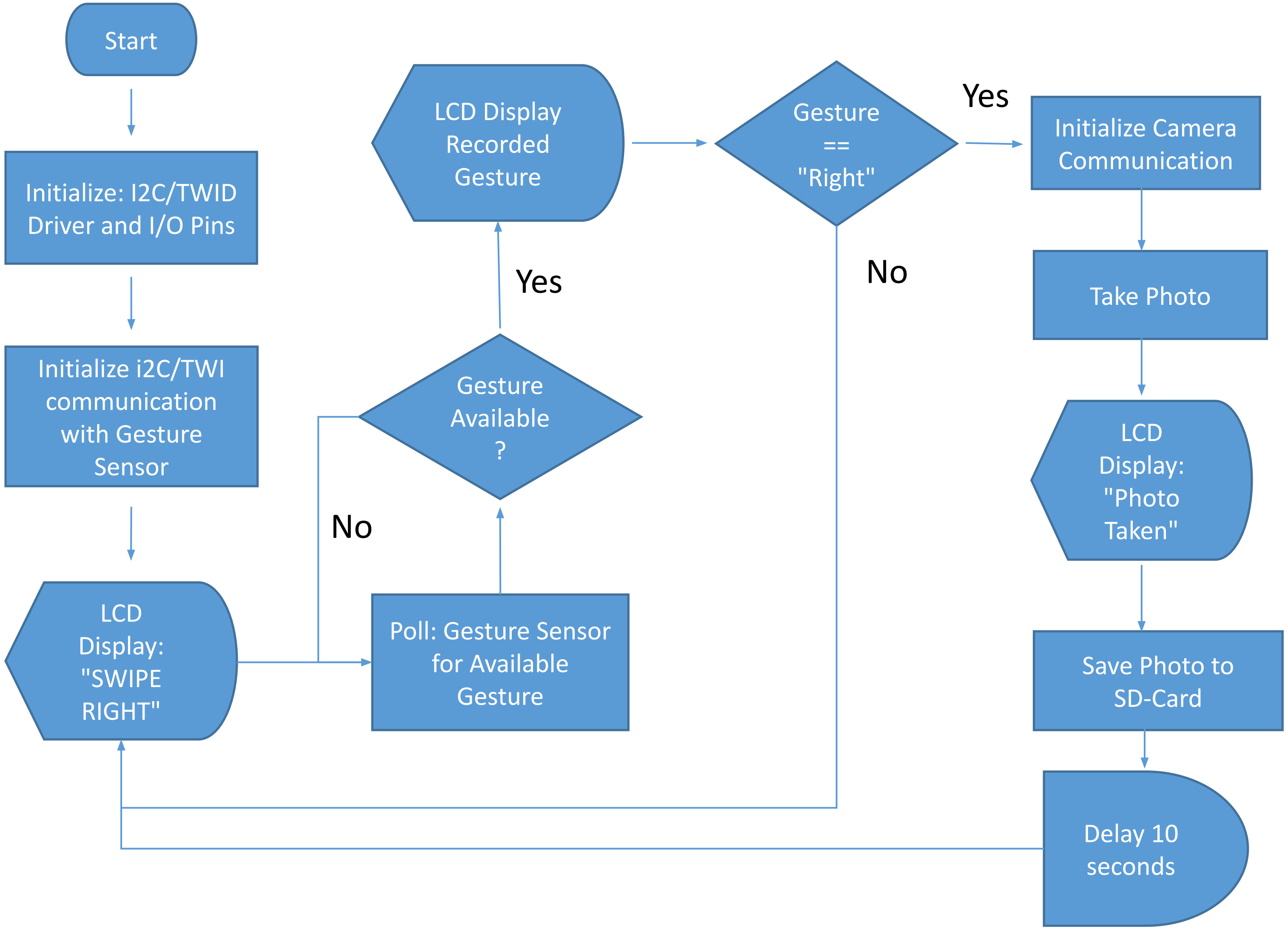
Computer: Receive and display image files

Human Supplied Gestures

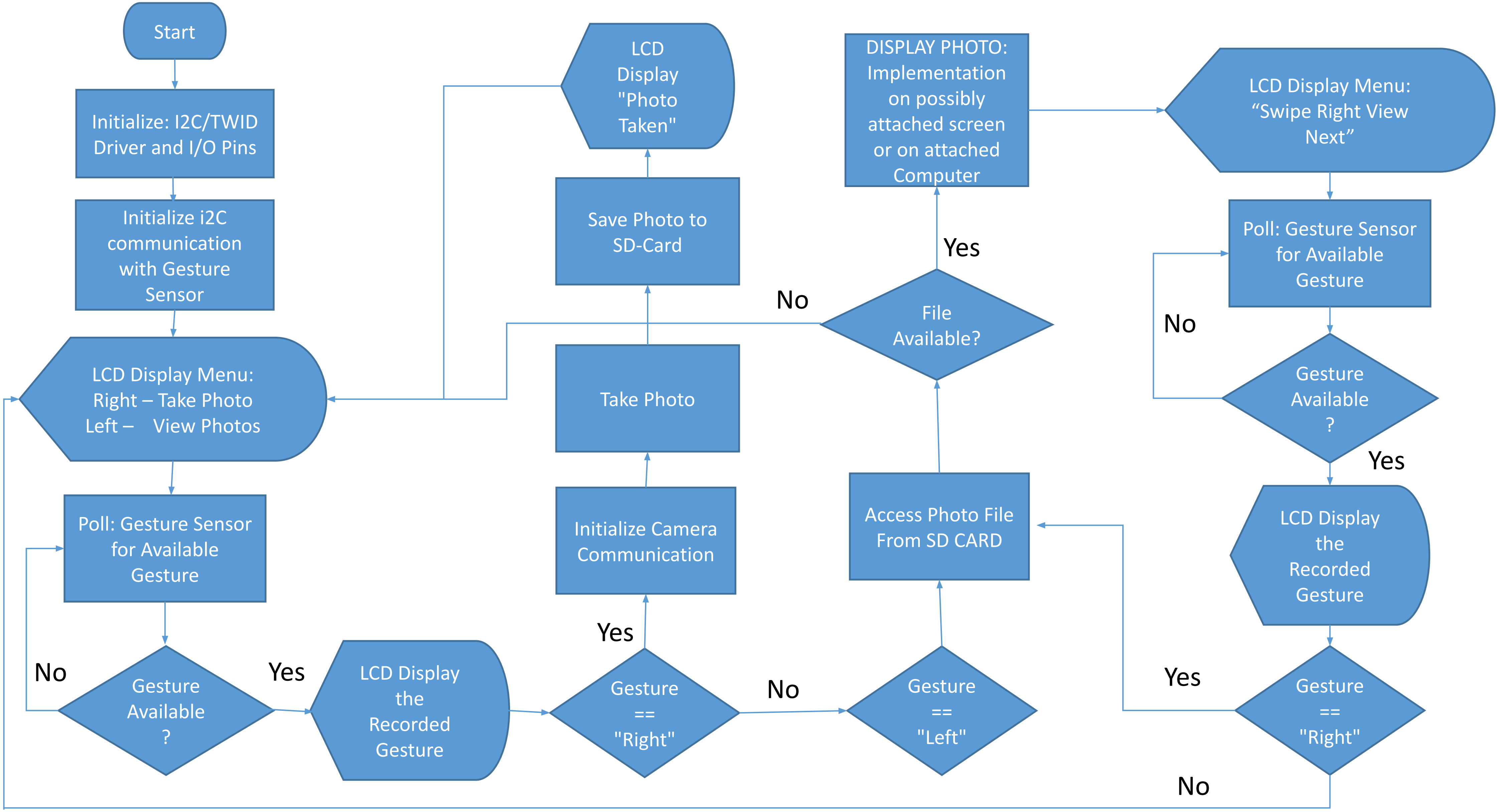
ATMEL AVR4019: Light and Proximity One

SDCARD

Program Flow (Basic):


Start
Initialize: I2C/TWID Driver and I/O Pins
Initialize i2C/TWI communication with Gesture Sensor
LCD Display: "SWIPE RIGHT"
Poll: Gesture Sensor for Available Gesture
Gesture Available ?
No
Yes
LCD Display Recorded Gesture
Gesture == "Right"
No
Yes
Initialize Camera Communication
Take Photo
LCD Display: "Photo Taken"
Save Photo to SD-Card
Delay 10 seconds

Program Flow (Advanced):


Start
Initialize: I2C/TWID Driver and I/O Pins
Initialize i2C communication with Gesture Sensor
LCD Display Menu: Right – Take Photo Left – View Photos
Poll: Gesture Sensor for Available Gesture
Gesture Available ?
No
Yes
LCD Display the Recorded Gesture
Gesture == "Right"
Yes
Initialize Camera Communication
Take Photo
Save Photo to SD-Card
LCD Display "Photo Taken"
No
Gesture == "Left"
Access Photo File From SD CARD
File Available?
No
Yes
DISPLAY PHOTO: Implementation on possibly attached screen or on attached Computer
LCD Display Menu: "Swipe Right View Next"
Poll: Gesture Sensor for Available Gesture
Gesture Available ?
No
Yes
LCD Display the Recorded Gesture
Gesture == "Right"
Yes
No

Example of a Gesture Sensor in use:

This is a Sparkfun Sensor not the Atmel.

Sparkfun Electronics. "SparkFun RGB and Gesture Sensor"." YouTube. Sparkfun, November , 2014. Video. 20 April 2016.

How does the SAMv71 communicate with the AVR4019 Sensor?

ATMEL Gesture sensor uses Phillips I^2C short for standard Inter-IC (integrated circuit) bus

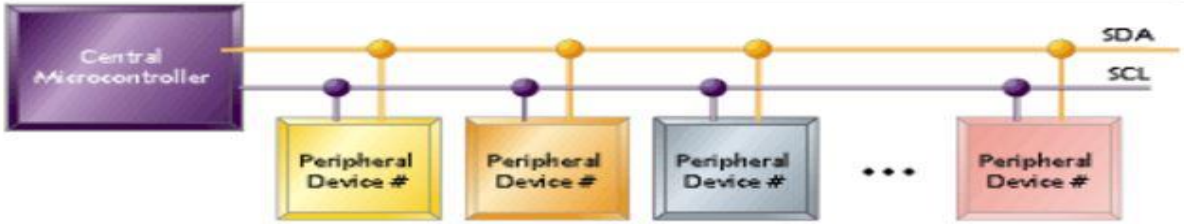
SAMv71 implements the bus under the name Twin-wire High Speed- Interface (TWIHS)

I^2C uses two lines: Serial Data (SDA) and Serial Clock (SCL)
Two types of devices: Masters and Slaves
Master - The device which initiates a transfer, generates clock signals and terminates a transfer.
Slave   - Device addressed by Master
All devices can send and receive, however, only master can tell the slave when to transmit data back.

High to Low transition on SDA line while SCL is High = START Condition

Low to High transition on SDA line while SCL is high = STOP Condition

Figure 1: I²C has two lines in total



David Kalinsky and Roee Kalinsky [Graphic]. Retrieved from
http://www.embedded.com/electronics-blogs/beginner-s-corner/4023816/Introduction-to-I2C

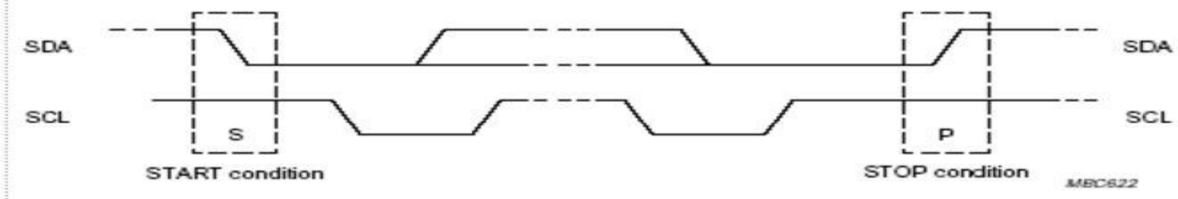
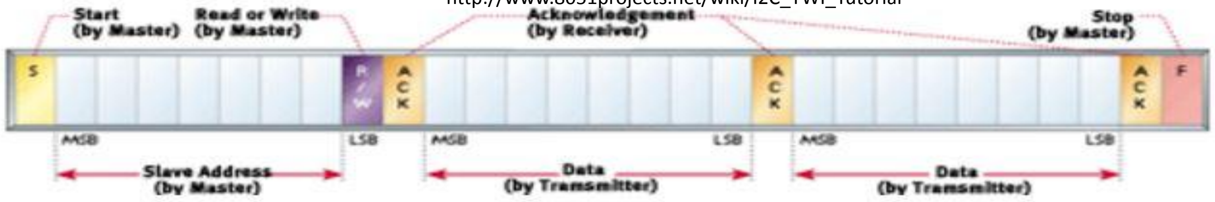


Fig.5 START and STOP conditions.

No Author Listed, [Graphic], retrieved from http://www.8051projects.net/wiki/I2C_TWI_Tutorial

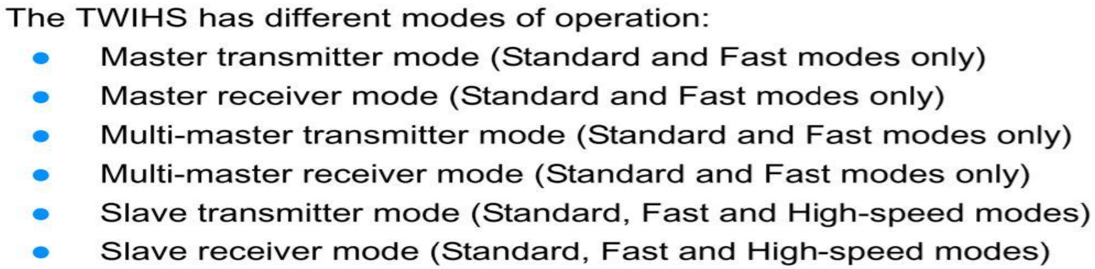
Figure 2: I²C communication

http://www.8051projects.net/wiki/I2C_TWI_Tutorial



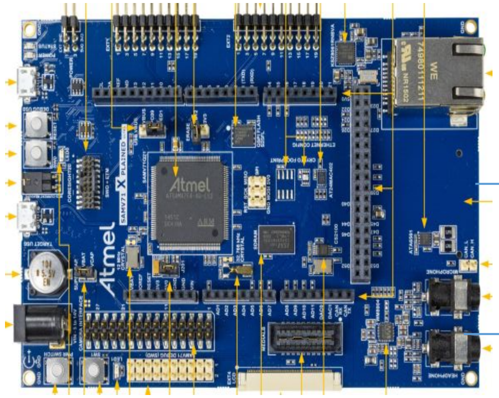
David Kalinsky and Roee Kalinsky [Graphic]. Retrieved from
http://www.embedded.com/electronics-blogs/beginner-s-corner/4023816/Introduction-to-I2C

SAMv71 to Gesture Sensor Communication:



The TWIHS has different modes of operation:

- Master transmitter mode (Standard and Fast modes only)
- Master receiver mode (Standard and Fast modes only)
- Multi-master transmitter mode (Standard and Fast modes only)
- Multi-master receiver mode (Standard and Fast modes only)
- Slave transmitter mode (Standard, Fast and High-speed modes)
- Slave receiver mode (Standard, Fast and High-speed modes)



GRND

VCC 3.3 V

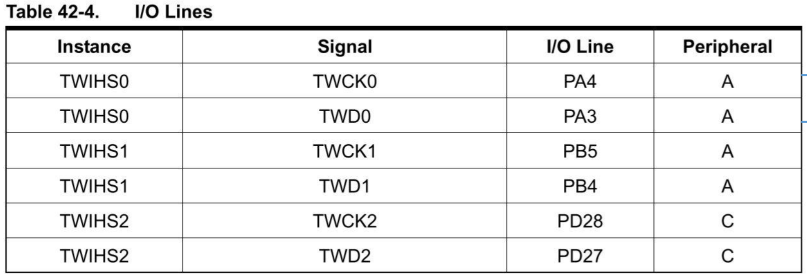


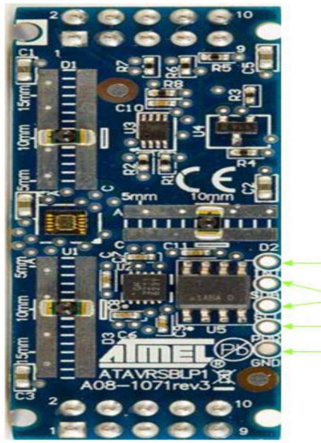
Table 42-4. I/O Lines

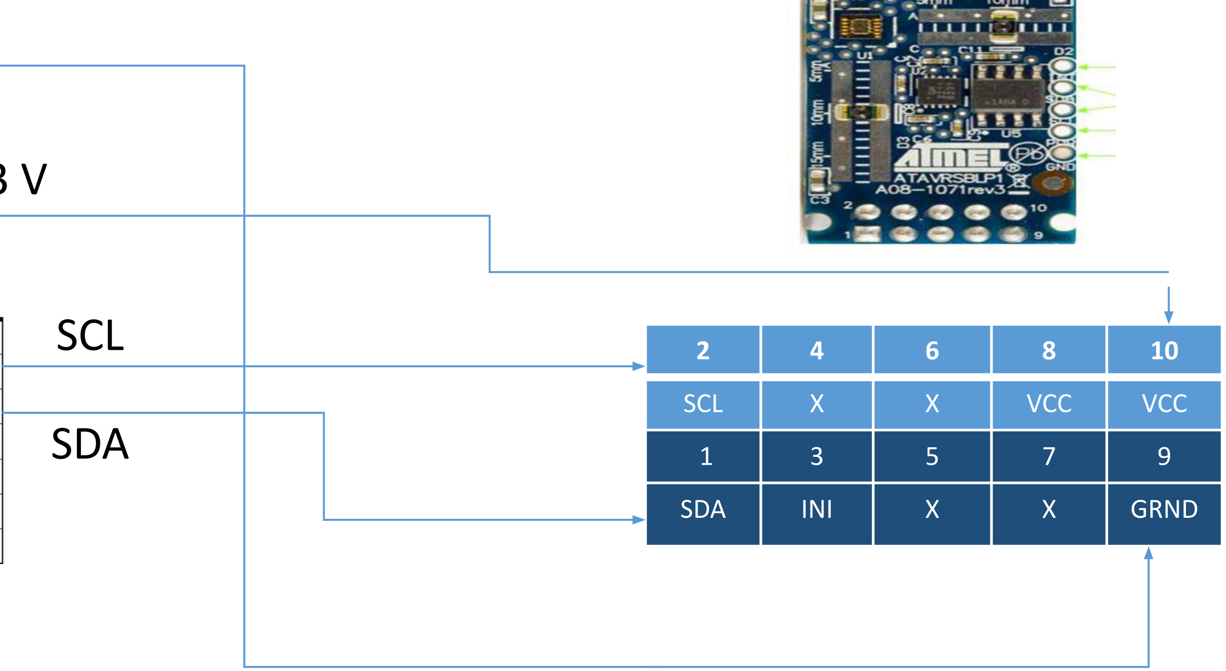
Instance	Signal	I/O Line	Peripheral
TWIHS0	TWCK0	PA4	A
TWIHS0	TWD0	PA3	A
TWIHS1	TWCK1	PB5	A
TWIHS1	TWD1	PB4	A
TWIHS2	TWCK2	PD28	C
TWIHS2	TWD2	PD27	C

SCL

SDA

Slave Address: 0x38





2	4	6	8	10
SCL	X	X	VCC	VCC
1	3	5	7	9
SDA	INI	X	X	GRND

SAMv71 TWIHS Implementation:

Registers used to implement TWIHS on the SAMv71

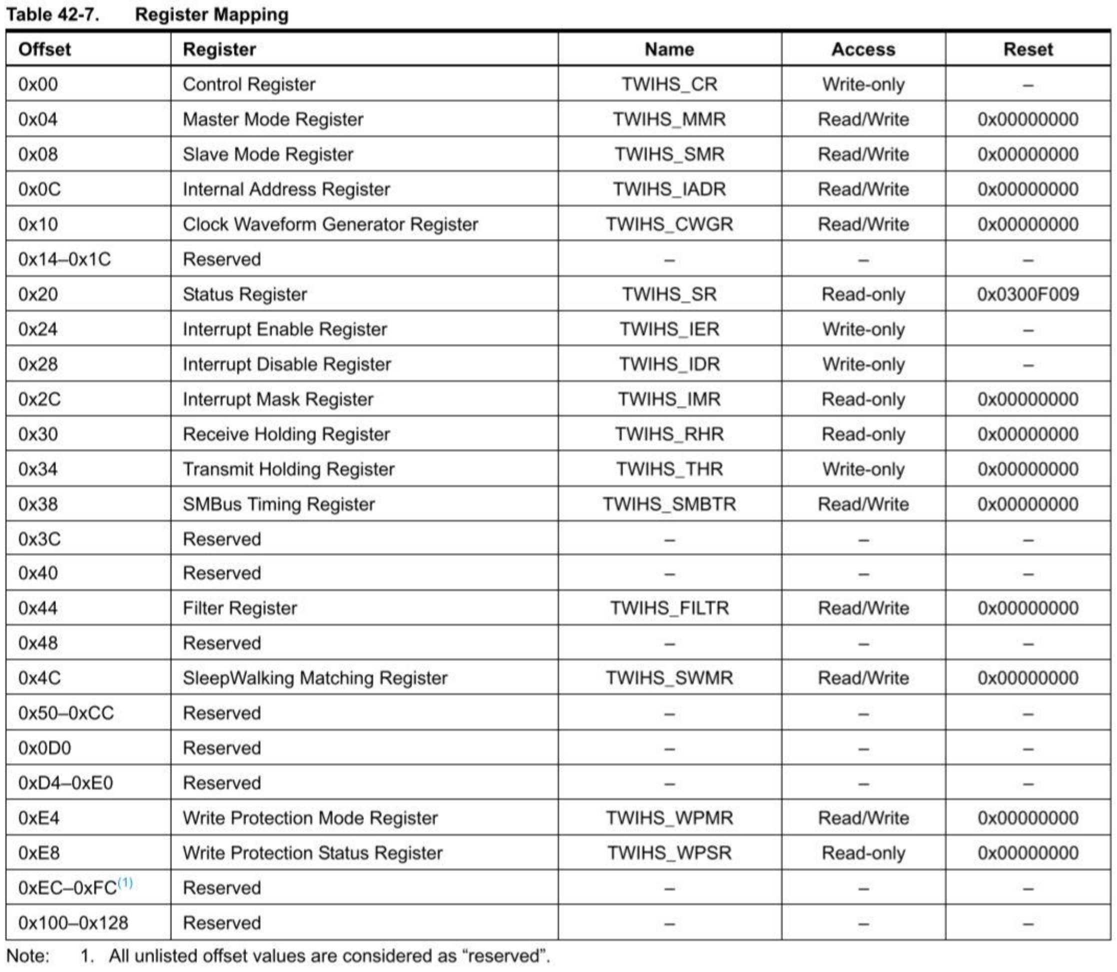
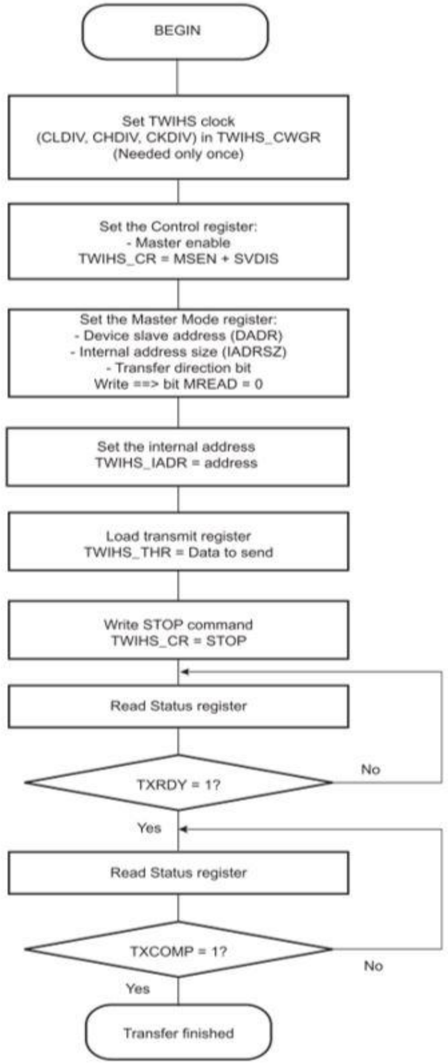


Table 42-7. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	TWIHS_CR	Write-only	–
0x04	Master Mode Register	TWIHS_MMR	Read/Write	0x00000000
0x08	Slave Mode Register	TWIHS_SMR	Read/Write	0x00000000
0x0C	Internal Address Register	TWIHS_IADR	Read/Write	0x00000000
0x10	Clock Waveform Generator Register	TWIHS_CWGR	Read/Write	0x00000000
0x14–0x1C	Reserved	–	–	–
0x20	Status Register	TWIHS_SR	Read-only	0x0300F009
0x24	Interrupt Enable Register	TWIHS_IER	Write-only	–
0x28	Interrupt Disable Register	TWIHS_IDR	Write-only	–
0x2C	Interrupt Mask Register	TWIHS_IMR	Read-only	0x00000000
0x30	Receive Holding Register	TWIHS_RHR	Read-only	0x00000000
0x34	Transmit Holding Register	TWIHS_THR	Write-only	0x00000000
0x38	SMBus Timing Register	TWIHS_SMBTR	Read/Write	0x00000000
0x3C	Reserved	–	–	–
0x40	Reserved	–	–	–
0x44	Filter Register	TWIHS_FILTR	Read/Write	0x00000000
0x48	Reserved	–	–	–
0x4C	SleepWalking Matching Register	TWIHS_SWMR	Read/Write	0x00000000
0x50–0xCC	Reserved	–	–	–
0x0D0	Reserved	–	–	–
0xD4–0xE0	Reserved	–	–	–
0xE4	Write Protection Mode Register	TWIHS_WPMR	Read/Write	0x00000000
0xE8	Write Protection Status Register	TWIHS_WPSR	Read-only	0x00000000
0xEC–0xFC[1]	Reserved	–	–	–
0x100–0x128	Reserved	–	–	–

Note: 1. All unlisted offset values are considered as "reserved".

TWIHS Write Operation with Single Data Byte and Internal Address



BEGIN
Set TWIHS clock (CLDIV, CHDIV, CKDIV) in TWIHS_CWGR (Needed only once)
Set the Control register: - Master enable TWIHS_CR = MSEN + SVDIS
Set the Master Mode register: - Device slave address (DADR) - Internal address size (IADRSZ) - Transfer direction bit Write ==> bit MREAD = 0
Set the internal address TWIHS_IADR = address
Load transmit register TWIHS_THR = Data to send
Write STOP command TWIHS_CR = STOP
Read Status register
TXRDY = 1?
No
Yes
Read Status register
TXCOMP = 1?
No
Yes
Transfer finished

Atmel Corporation, "SMART SAM V71Q SAM V71N SAM V71J Preliminary Datasheet ." (2015): 999-1063. Web <http://www.atmel.com/images/atmel-44003-32-bit-cortex-m7-microcontroller-sam-v71q-sam-v71n-sam-v71j_datasheet.pdf>

SAMv71 TWIHS :

twi.h – Interface for configuration of Twin Wire Interface

twid.h – TWI driver Interface.
Builds upon twi.h contains functions to read / write across TWI interface.

TWID_Read(Pointer to TWI, Slave Address, Internal Slave Address, Received Bytes Stg, # of Bytes to receive, Asynch);

Example: TWID_Read(&twid, 0x10, 0x04, 1, Read_Data, 1, 0);

TWID_Write(&twid, Slave Address,Internal Slave Address (Optional),# of internal address bytes, 1, Buffer for Storing Received Data, Data buffer to Send, Asynch);

Example: TWID_Write(&twid, 0x10, 0x05, 1, &Write_Response ,0x03, 0);

ATMEL AVR4019 Light and Proximity Sensor:

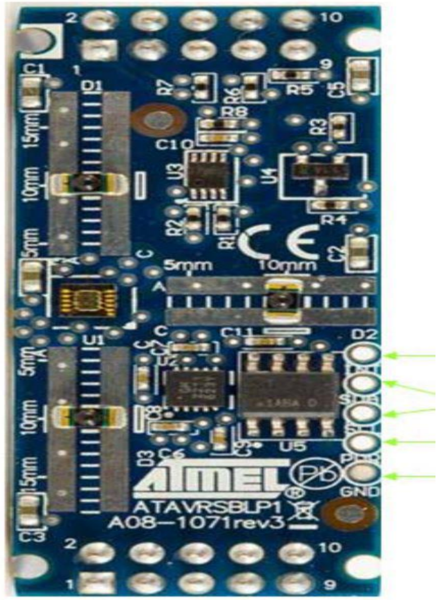
Some issues with Supplied documentation:

Lacked discussion of registers and how to use them.

Focused on Sensors Xplained Application Programming Interface (API)

I could not get the API to work for the SAMv71:

Almost all of the documentation focused on ATMEL STUDIO.
Issues finding device.h support for this specific sensor.



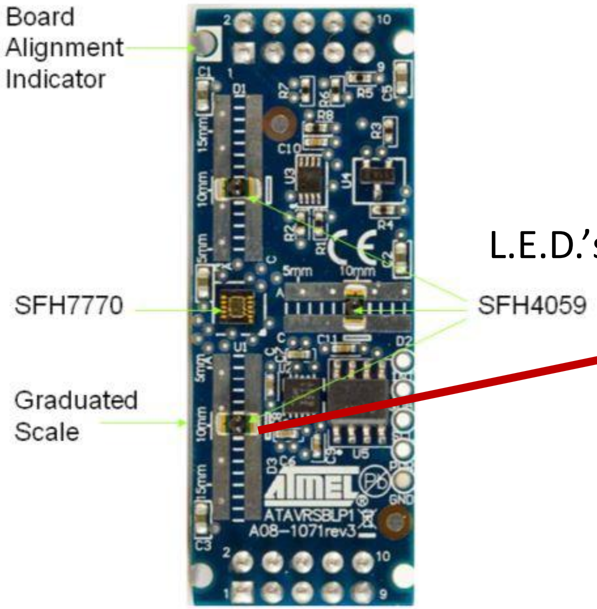
Atmel Corporation, "SMART SAM V71Q SAM V71N SAM V71J Preliminary Datasheet ." (2015): 999-1063. Web <http://www.atmel.com/images/atmel-44003-32-bit-cortex-m7-microcontroller-sam-v71q-sam-v71n-sam-v71j_datasheet.pdf>

ATMEL AVR4019 Light and Proximity Sensor:

OSRAM: SFH7770 Ambient Light and Proximity Sensor – Mounted on the Atmel AVR4019

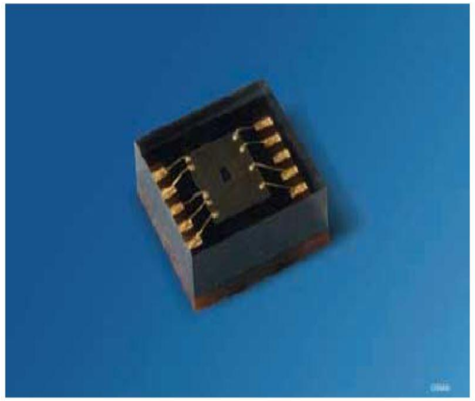
OSRAM: SF7770 much more detailed documentation provided.

Figure 4-1. Sensor arrangement.



L.E.D.'s

OSRAM: Ambient Light and Proximity Sensor SFH 7770



Atmel Corporation, "Atmel AVR4019: Light and Proximity One (ATAVRSBLP1) Hardware Users Guide ." (2011): 1-7. Web <http://www.atmel.com/images/doc8471.pdf>

Dr. Bernhard Stojetz, "Ambient Light and Proximity Sensor SFH 7770 Application Note ." (2011): 1-28. Web <http://www.osram-os.com/Graphics/XPic9/00055496_0.pdf/Proximity%20Sensor%20SFH%207770.pdf>

ATMEL AVR4019 Light and Proximity Sensor:

OSRAM: SFH7770 Ambient Light and Proximity Sensor – Mounted on the Atmel AVR4019

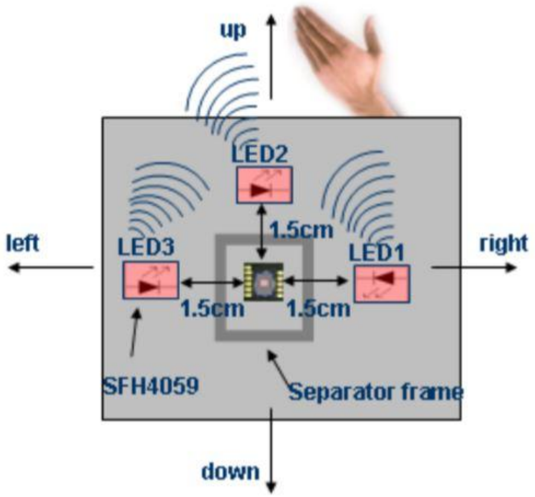


Figure 14: Sample setup for gesture recognition with SFH7770, each containing 3 LEDs

All three L.E.D.'s equally spaced from the detector 10 mm
Atmel Board

Open hand is moved a few Centimeters above the sensor

The LED signal reading on the proximity sensor decreases as the distance from the object increases.

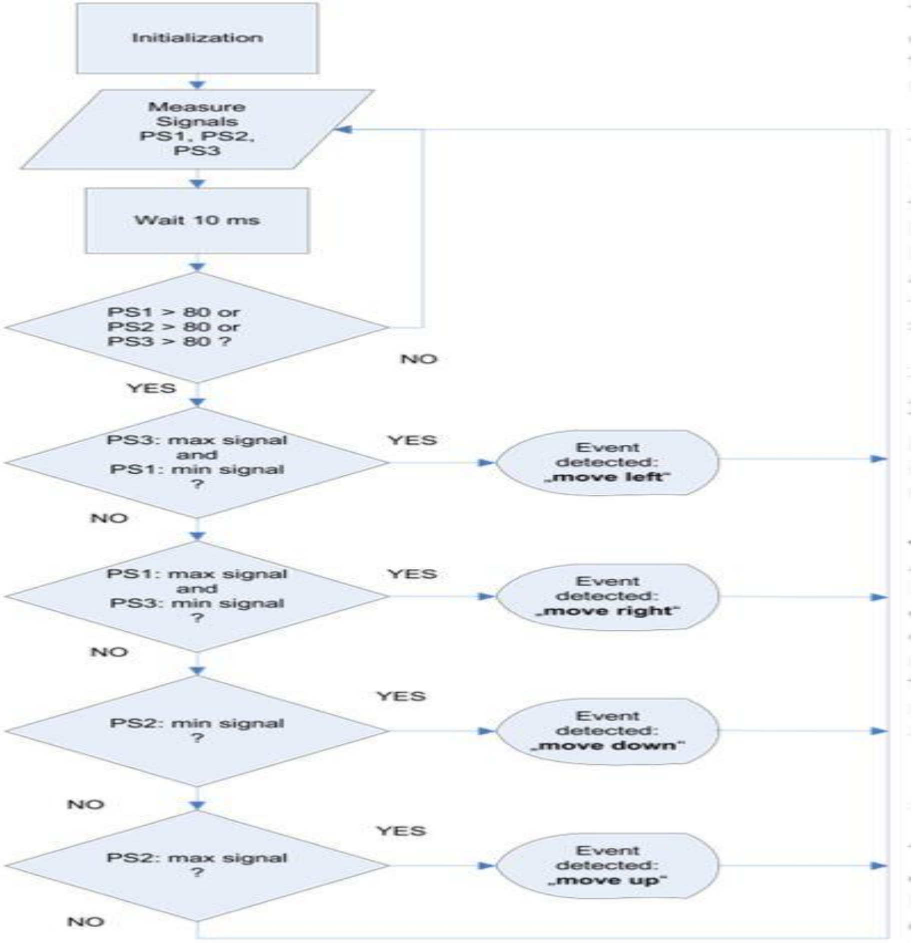
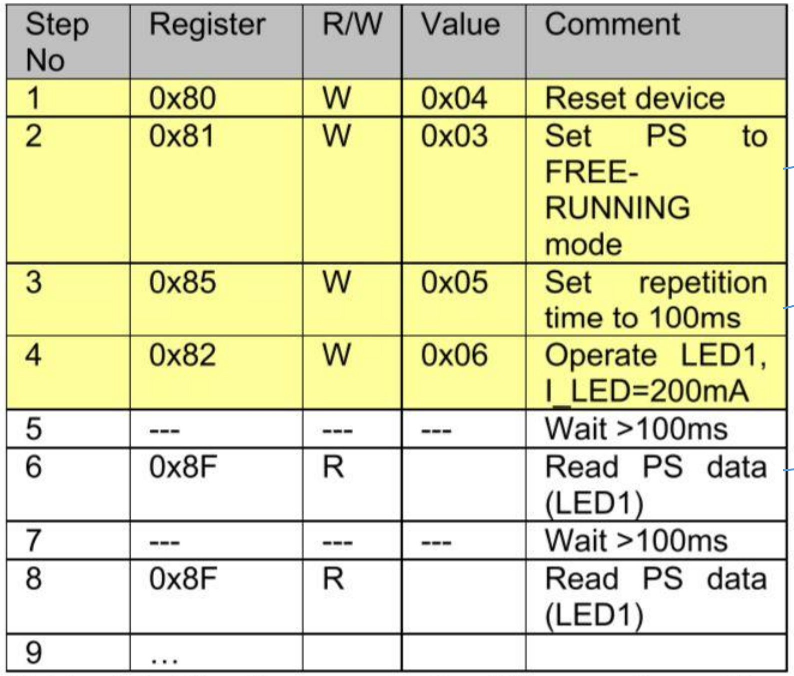

Initialization
Measure Signals PS1, PS2, PS3
Wait 10 ms
PS1 > 80 or PS2 > 80 or PS3 > 80 ?
NO
YES
PS3: max signal and PS1: min signal ?
YES
Event detected: „move left"
NO
PS1: max signal and PS3: min signal ?
YES
Event detected: „move right"
NO
PS2: min signal ?
YES
Event detected: „move down"
NO
PS2: max signal ?
YES
Event detected: „move up"
NO

Figure 15: Gesture recognition sample flow for SFH7770 and 3 LEDs

Dr. Bernhard Stojetz, "Ambient Light and Proximity Sensor SFH 7770 Application Note ." (2011): 1-28. Web <http://www.osram-os.com/Graphics/XPic9/00055496_0.pdf/Proximity%20Sensor%20SFH%207770.pdf>

ATMEL AVR4019 Light and Proximity Sensor:

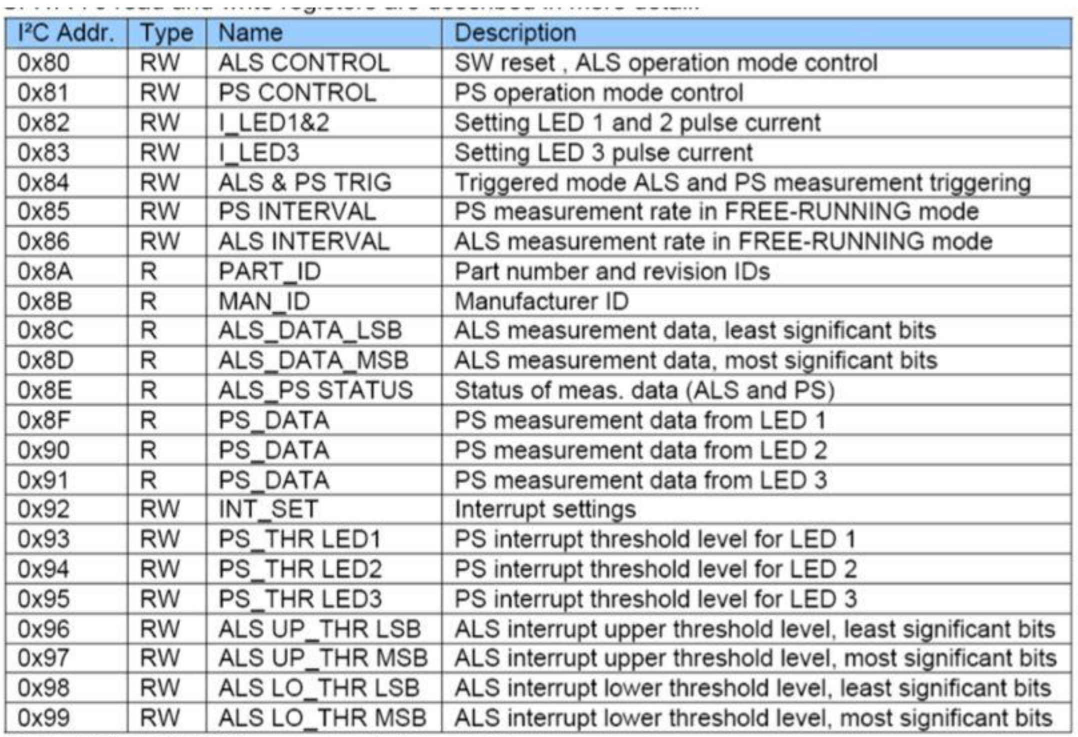
How to initiate communication with Gesture Sensor?



Step No	Register	R/W	Value	Comment
1	0x80	W	0x04	Reset device
2	0x81	W	0x03	Set PS to FREE-RUNNING mode
3	0x85	W	0x05	Set repetition time to 100ms
4	0x82	W	0x06	Operate LED1, I_LED=200mA
5	---	---	---	Wait >100ms
6	0x8F	R		Read PS data (LED1)
7	---	---	---	Wait >100ms
8	0x8F	R		Read PS data (LED1)
9	…			

Table 4-1: Simple program for PS operation with single LED (LED1)

SFH7770 Gesture Sensor Registers



I²C Addr.	Type	Name	Description
0x80	RW	ALS CONTROL	SW reset , ALS operation mode control
0x81	RW	PS CONTROL	PS operation mode control
0x82	RW	I_LED1&2	Setting LED 1 and 2 pulse current
0x83	RW	I_LED3	Setting LED 3 pulse current
0x84	RW	ALS & PS TRIG	Triggered mode ALS and PS measurement triggering
0x85	RW	PS INTERVAL	PS measurement rate in FREE-RUNNING mode
0x86	RW	ALS INTERVAL	ALS measurement rate in FREE-RUNNING mode
0x8A	R	PART_ID	Part number and revision IDs
0x8B	R	MAN_ID	Manufacturer ID
0x8C	R	ALS_DATA_LSB	ALS measurement data, least significant bits
0x8D	R	ALS_DATA_MSB	ALS measurement data, most significant bits
0x8E	R	ALS_PS STATUS	Status of meas. data (ALS and PS)
0x8F	R	PS_DATA	PS measurement data from LED 1
0x90	R	PS_DATA	PS measurement data from LED 2
0x91	R	PS_DATA	PS measurement data from LED 3
0x92	RW	INT_SET	Interrupt settings
0x93	RW	PS_THR LED1	PS interrupt threshold level for LED 1
0x94	RW	PS_THR LED2	PS interrupt threshold level for LED 2
0x95	RW	PS_THR LED3	PS interrupt threshold level for LED 3
0x96	RW	ALS UP_THR LSB	ALS interrupt upper threshold level, least significant bits
0x97	RW	ALS UP_THR MSB	ALS interrupt upper threshold level, most significant bits
0x98	RW	ALS LO_THR LSB	ALS interrupt lower threshold level, least significant bits
0x99	RW	ALS LO_THR MSB	ALS interrupt lower threshold level, most significant bits

Table 3-1: SFH7770 control and data registers

Dr. Bernhard Stojetz, "Ambient Light and Proximity Sensor SFH 7770 Application Note ." (2011): 1-28. Web <http://www.osram-os.com/Graphics/XPic9/00055496_0.pdf/Proximity%20Sensor%20SFH%207770.pdf>

Hardware Issues:

My progress was slowed with the gesture sensor due to hardware issues.

The SAM V71 board started to give errors when attempted to flash it.
Uninstalled and Reinstalled Keil, drivers, etc.
The board was falling on other computers also

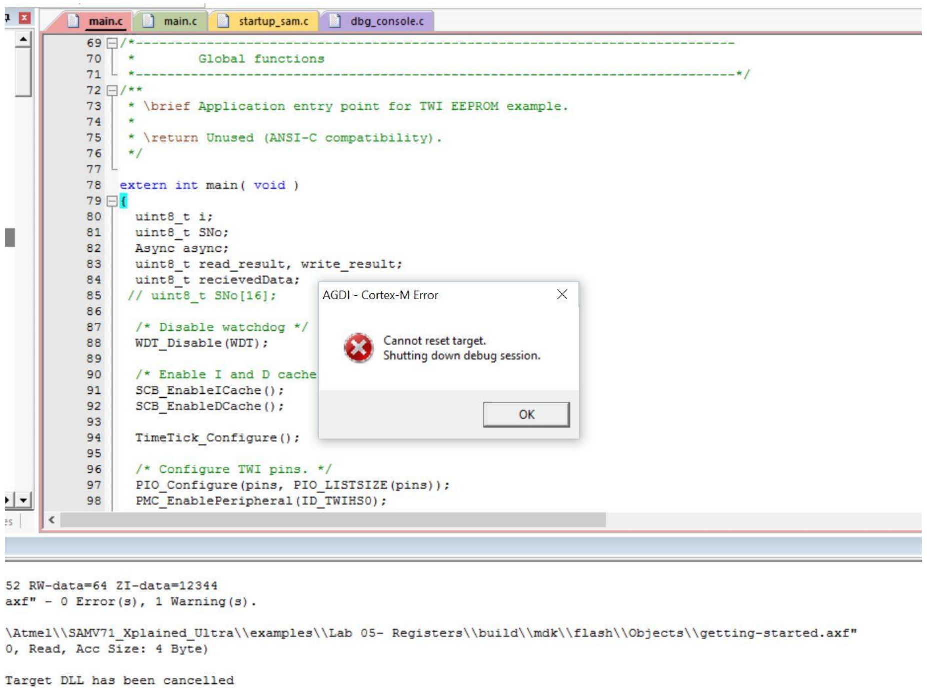
I purchased another board (simpler SAM E70)
It worked at first and then suddenly the same problems arose.

At this point I determined that it may be laptop?

I sent a detailed E-mail to ATMEL with error explanations and I have not heard anything back yet.

Using another laptop and the Professors board I have been able to continue working.

Most Recent Update: It appears that the problems I am having may be due to the firmware.

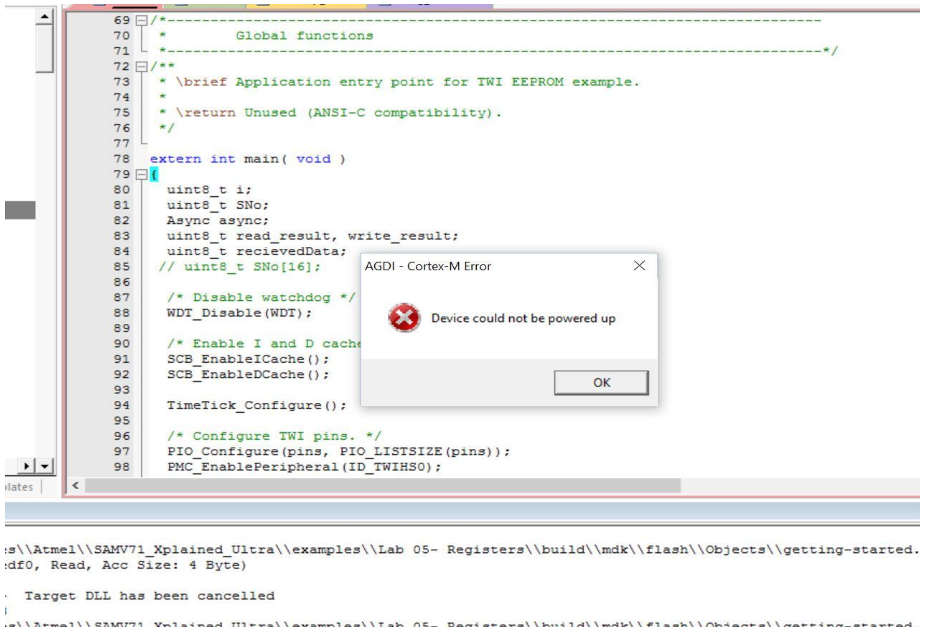


```
/*-----------------------------------------------------------------------------
 *         Global functions
 *---------------------------------------------------------------------------*/
/**
 * \brief Application entry point for TWI EEPROM example.
 *
 * \return Unused (ANSI-C compatibility).
 */

extern int main( void )
{
  uint8_t i;
  uint8_t SNo;
  Async async;
  uint8_t read_result, write_result;
  uint8_t recievedData;
 // uint8_t SNo[16];

  /* Disable watchdog */
  WDT_Disable(WDT);

  /* Enable I and D cache
  SCB_EnableICache();
  SCB_EnableDCache();

  TimeTick_Configure();

  /* Configure TWI pins. */
  PIO_Configure(pins, PIO_LISTSIZE(pins));
  PMC_EnablePeripheral(ID_TWIHS0);
```

AGDI - Cortex-M Error

Cannot reset target.
Shutting down debug session.

```
52 RW-data=64 ZI-data=12344
axf" - 0 Error(s), 1 Warning(s).

\Atmel\\SAMV71_Xplained_Ultra\\examples\\Lab 05- Registers\\build\\mdk\\flash\\Objects\\getting-started.axf"
0, Read, Acc Size: 4 Byte)

Target DLL has been cancelled
```



```
/*-----------------------------------------------------------------------------
 *         Global functions
 *---------------------------------------------------------------------------*/
/**
 * \brief Application entry point for TWI EEPROM example.
 *
 * \return Unused (ANSI-C compatibility).
 */

extern int main( void )
{
  uint8_t i;
  uint8_t SNo;
  Async async;
  uint8_t read_result, write_result;
  uint8_t recievedData;
 // uint8_t SNo[16];

  /* Disable watchdog */
  WDT_Disable(WDT);

  /* Enable I and D cache
  SCB_EnableICache();
  SCB_EnableDCache();

  TimeTick_Configure();

  /* Configure TWI pins. */
  PIO_Configure(pins, PIO_LISTSIZE(pins));
  PMC_EnablePeripheral(ID_TWIHS0);
```

AGDI - Cortex-M Error

Device could not be powered up

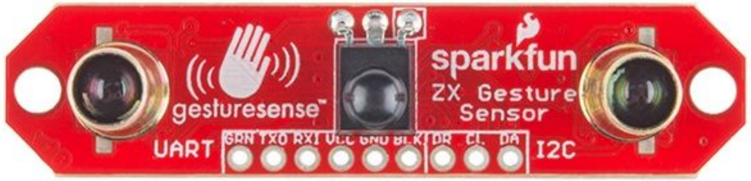
```
s\\Atmel\\SAMV71_Xplained_Ultra\\examples\\Lab 05- Registers\\build\\mdk\\flash\\Objects\\getting-started.
df0, Read, Acc Size: 4 Byte)

Target DLL has been cancelled

s\\Atmel\\SAMV71_Xplained_Ultra\\examples\\Lab 05- Registers\\build\\mdk\\flash\\Objects\\getting-started
```

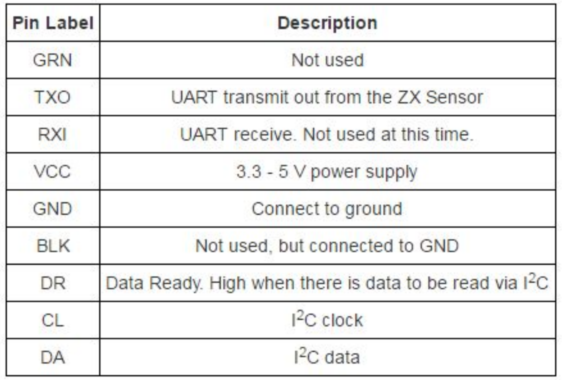
Plan B:

ATMEL gesture sensor was not responding.
I confirmed that the output pins were configured and data was being sent.

I purchased a Sparkfun Gesture Sensor earlier in the semester so I started to work with that.

ZX Distance and Gesture Sensor



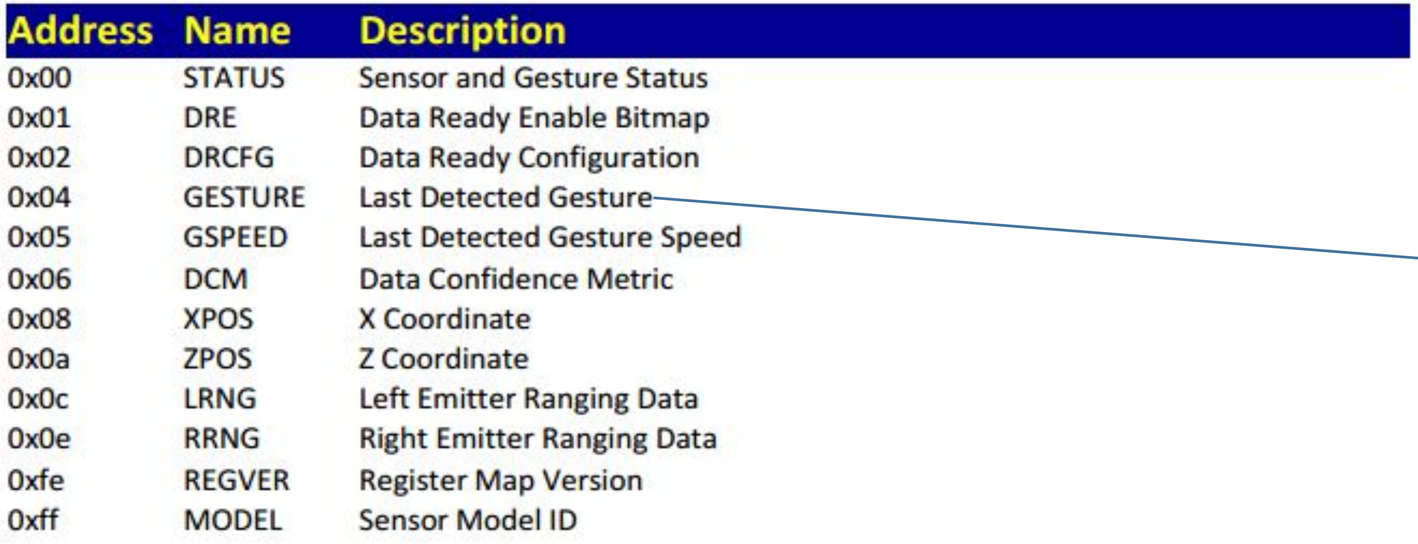


Pin Label	Description
GRN	Not used
TXO	UART transmit out from the ZX Sensor
RXI	UART receive. Not used at this time.
VCC	3.3 - 5 V power supply
GND	Connect to ground
BLK	Not used, but connected to GND
DR	Data Ready. High when there is data to be read via I^{2}C
CL	I^{2}C clock
DA	I^{2}C data

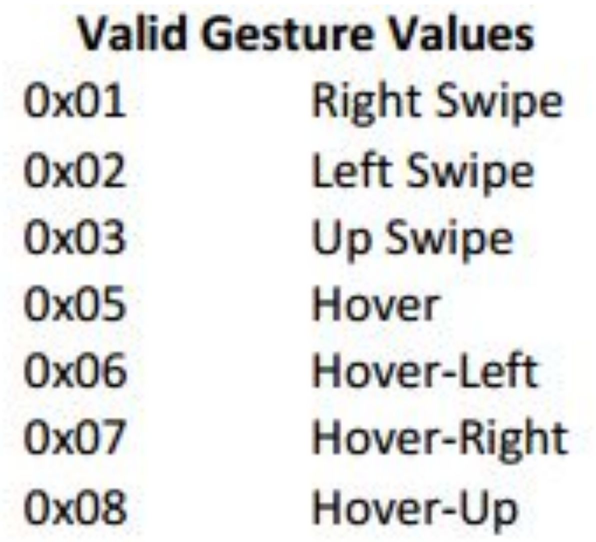
Hymel, Shawn, [Image & Graphic) Retrieved from https://learn.sparkfun.com/tutorials/zx-distance-and-gesture-sensor-hookup-guide#hardware-hookup

ZX Distance and Gesture Sensor:

Using the TWID Read command I was able to read the LAST DETECTED GESTURE register.



Address	Name	Description
0x00	STATUS	Sensor and Gesture Status
0x01	DRE	Data Ready Enable Bitmap
0x02	DRCFG	Data Ready Configuration
0x04	GESTURE	Last Detected Gesture
0x05	GSPEED	Last Detected Gesture Speed
0x06	DCM	Data Confidence Metric
0x08	XPOS	X Coordinate
0x0a	ZPOS	Z Coordinate
0x0c	LRNG	Left Emitter Ranging Data
0x0e	RRNG	Right Emitter Ranging Data
0xfe	REGVER	Register Map Version
0xff	MODEL	Sensor Model ID



Valid Gesture Values

0x01	Right Swipe
0x02	Left Swipe
0x03	Up Swipe
0x05	Hover
0x06	Hover-Left
0x07	Hover-Right
0x08	Hover-Up

No Author Listed, "GestureSense XZ01 Sensor I2C Register Map ." (2014): 1-5. Web <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/XYZ%20I2C%20Registers%20v1.zip>

Future Work on this Project:

Establish communication with ATMEL gesture sensor

.

Explore other features of both sensors I.e: Interrupts, sensitivity, accuracy

Integrate the other hardware components: Camera, LCD, etc.

References:

1. Atmel Corporation, "SMART SAM V71Q SAM V71N SAM V71J Preliminary Datasheet ." (2015): 999-1063. Web <http://www.atmel.com/images/atmel-44003-32-bit-cortex-m7-microcontroller-sam-v71q-sam-v71n-sam-v71j_datasheet.pdf>

2. Dr. Bernhard Stojetz, "Ambient Light and Proximity Sensor SFH 7770 Application Note ." (2011): 1-28. Web <http://www.osram-os.com/Graphics/XPic9/00055496_0.pdf/Proximity%20Sensor%20SFH%207770.pdf>

3. No Author Listed, "GestureSense XZ01 Sensor I2C Register Map ." (2014): 1-5. Web <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/XYZ%20I2C%20Registers%20v1.zip>

4. Atmel Corporation, "Atmel AVR4019: Light and Proximity One (ATAVRSBLP1) Hardware Users Guide ." (2011): 1-7. Web <http://www.atmel.com/images/doc8471.pdf>

5. Kalinsky, David, and Roee Kalinsky. "Introduction to I2C." Embedded: Cracking the Code, 31 July 2011. Web. 20 Apr. 2016.

6. No Author Listed. "I^{2}C Explained." Electric Imp -. © 2011-2016 Electric Imp, Inc., 1 Jan. 2011. Web. 26 Apr. 2016.

7. Andrew Lohbihler. "XYZ Interactive Technologies, Using the ZX SparkFun sensor for Arduino." CTO, XYZ Interactive Technologies Inc., No Date. Web. 26 Apr. 2016. <XYZ Interactive Technologies, Using the ZX SparkFun sensor for Arduino>

Questions?