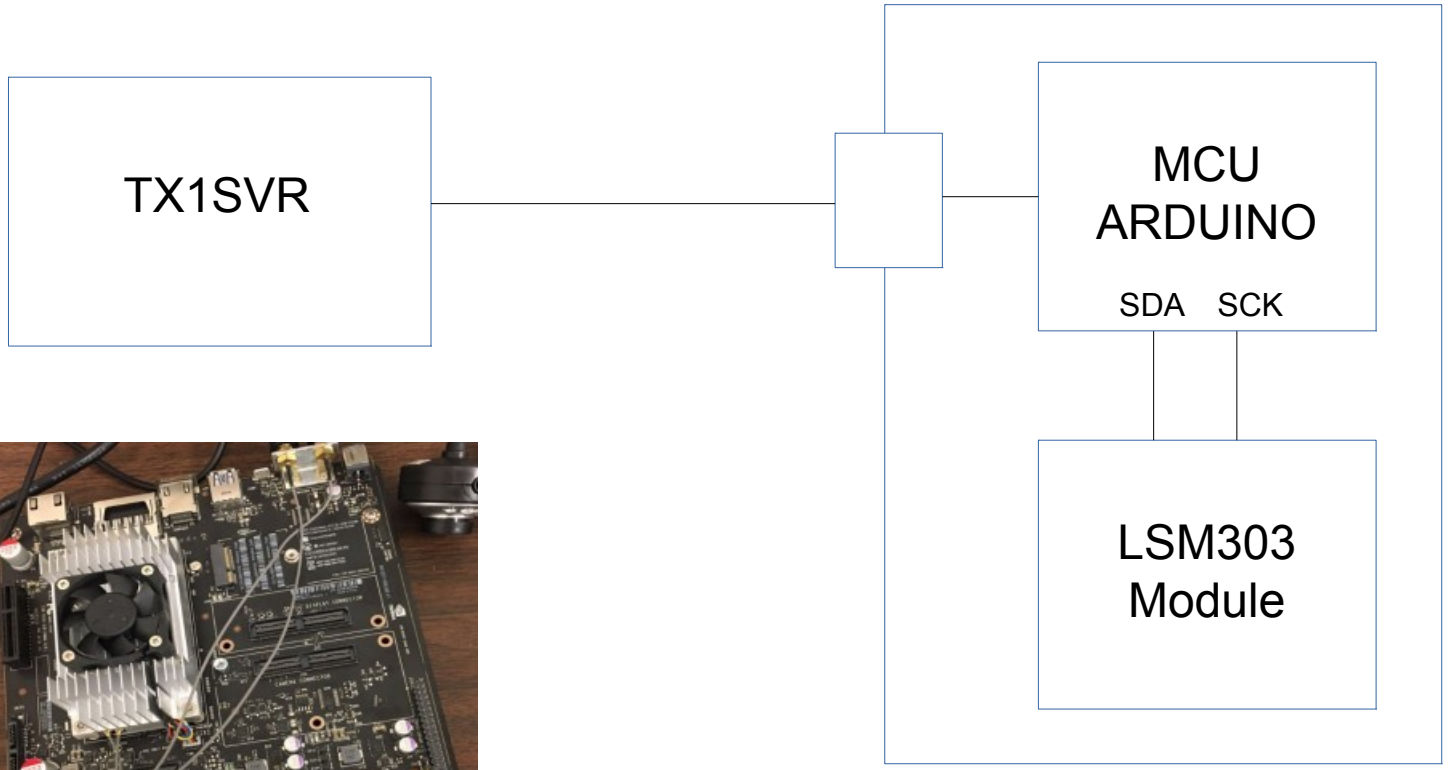
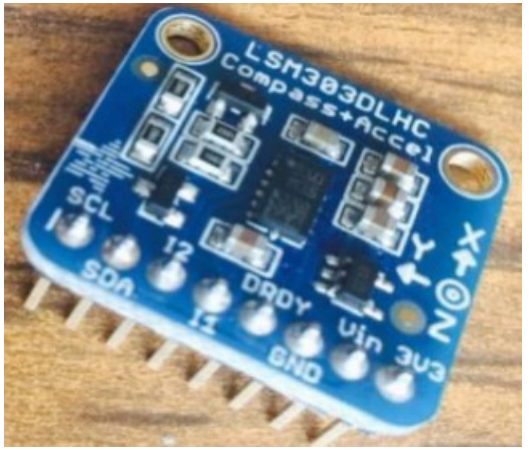
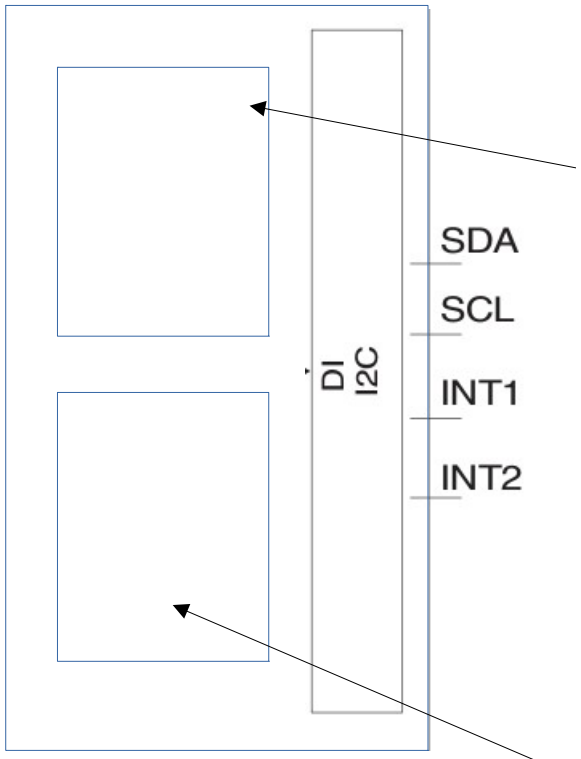
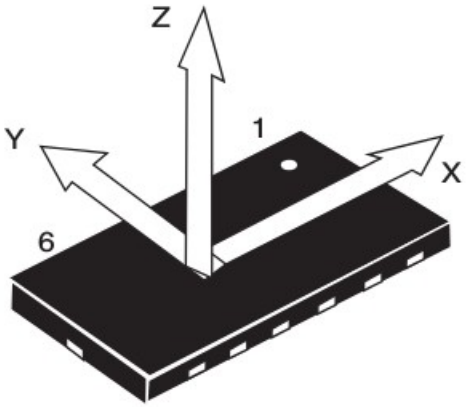
System Architecture for 3D HMI Device Design



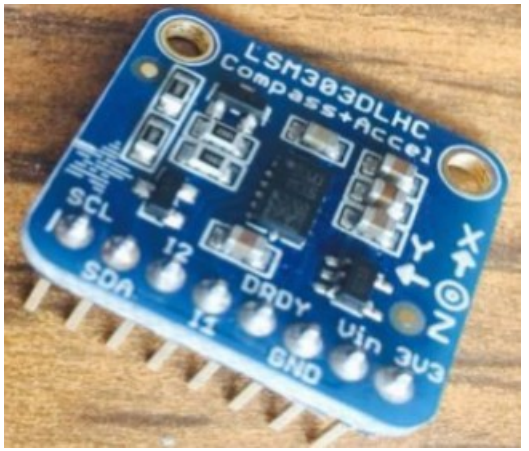


LSM303 Sensor and I2C Interface

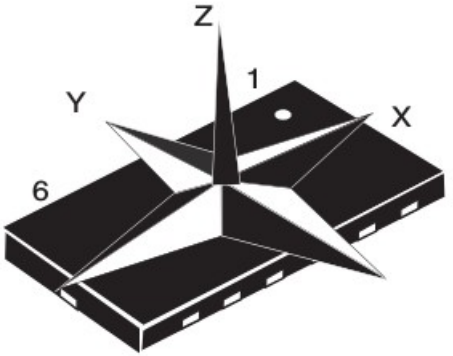




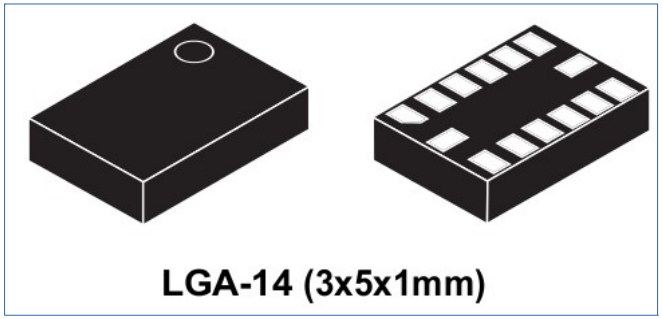
3D accelerometer



LSM303DLHC



3D magnetometer



Pin name	Pin description
SCL	I^2C serial clock (SCL)
SDA	I^2C serial data (SDA)

I2C Hardware Interface

Pull-up resistors (recommended value 10 kOhm) are placed on the two I2C bus lines.

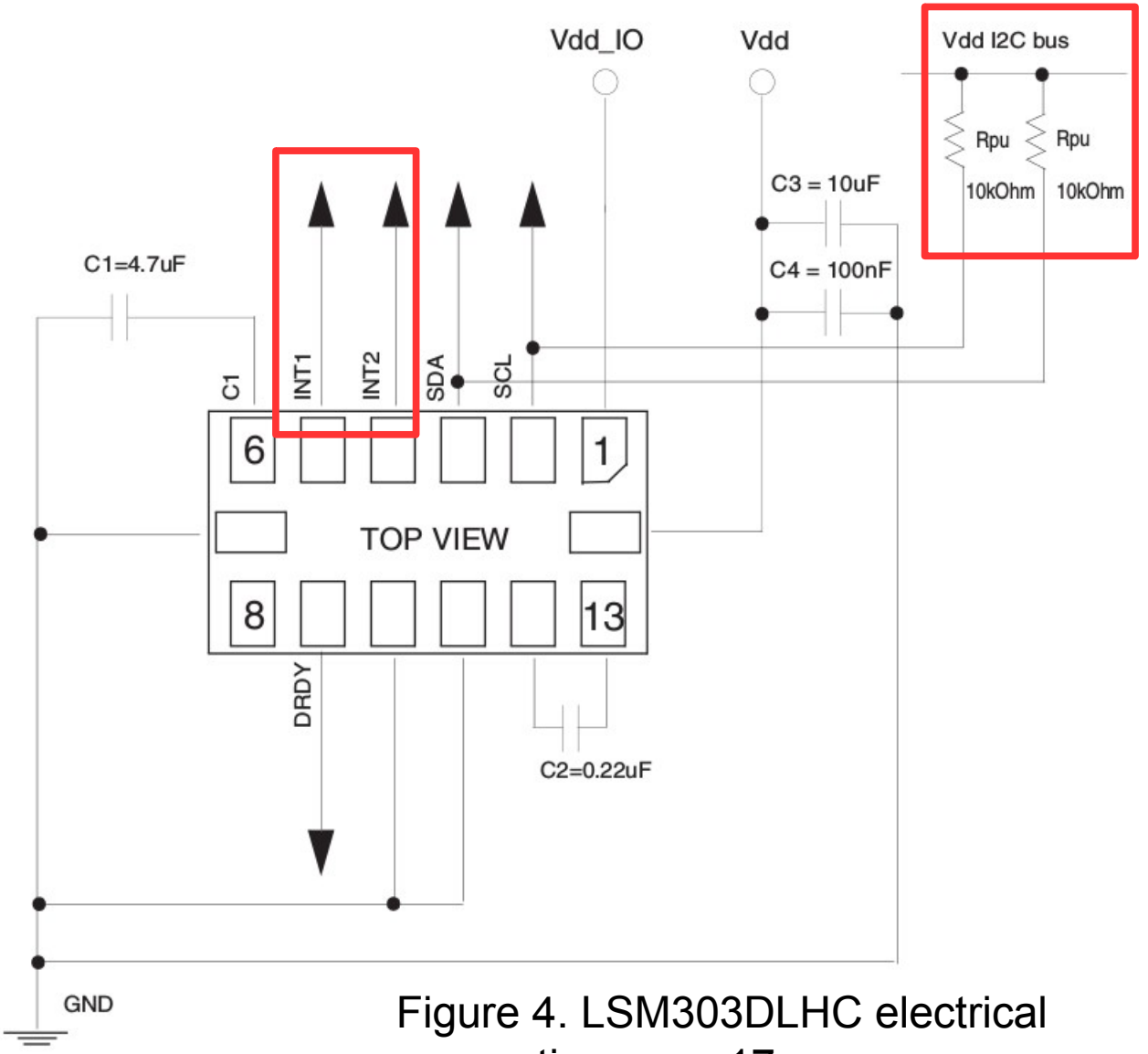
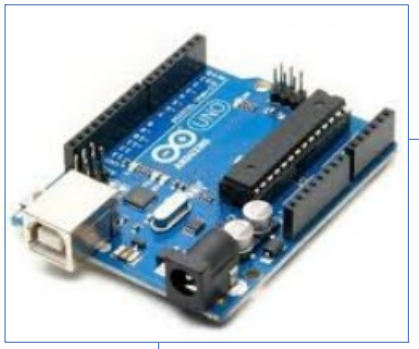
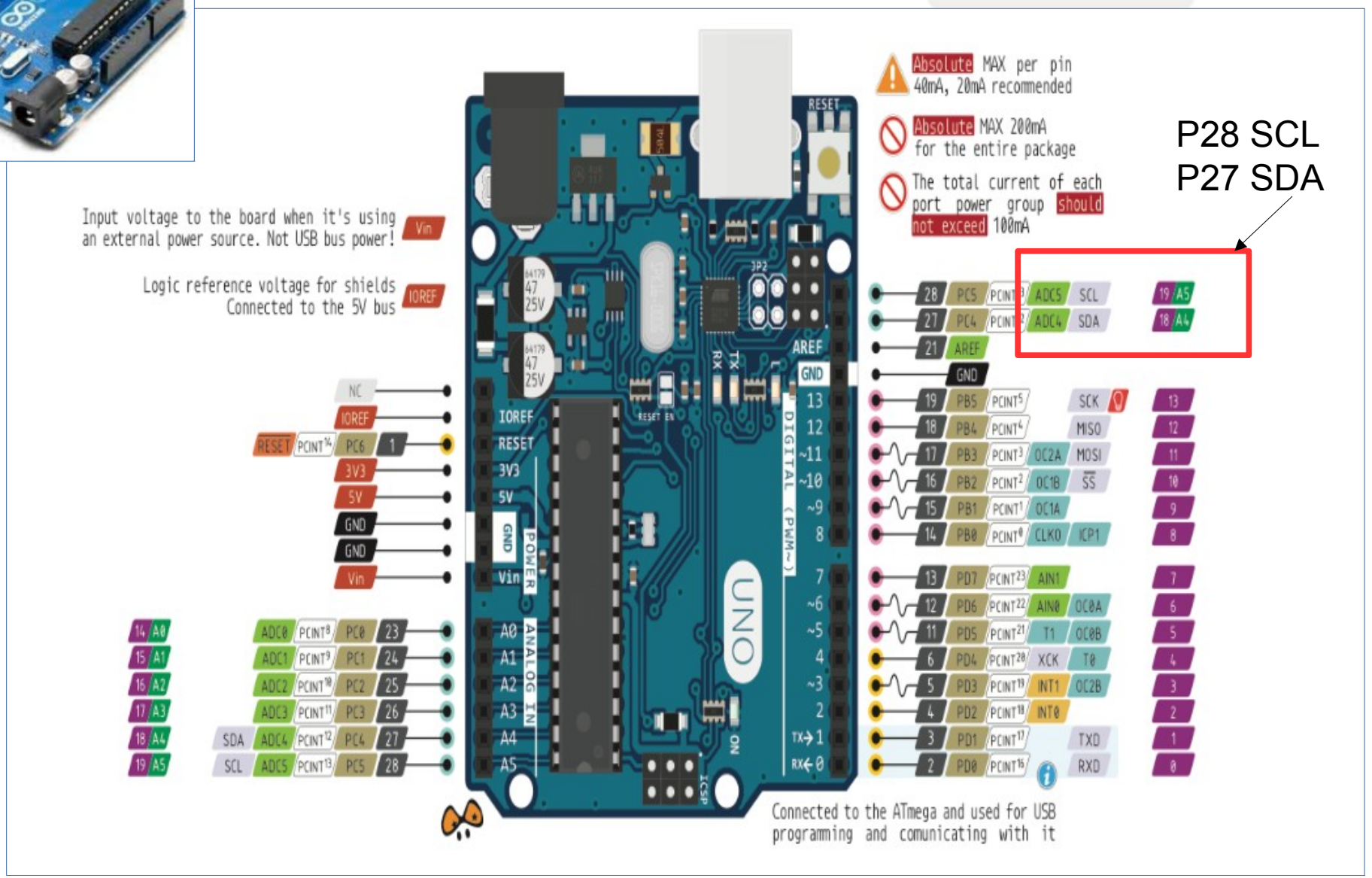


Figure 4. LSM303DLHC electrical connections, pp. 17

I2C On Arduino



http://forum.arduino.cc/index.php?topic=146315.0



I2C Timing Description

Table 11. Transfer when master is writing one byte to slave, pp. 20

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

Table 12. Transfer when master is writing multiple bytes to slave:

Master	ST	SAD + W		SUB		DATA		DATA		SP
Slave			SAK		SAK		SAK		SAK	

Table 13. Transfer when master is receiving (reading) one byte of data from slave

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

The transaction on the bus is started through a START (ST) signal. A START condition is defined as a high-to-low transition on the data line while the SCL line is held high. After this has been transmitted by the master, the bus is considered busy. The next byte of data transmitted after the start condition contains the address of the slave in the first 7 bits and bit 8 tells whether the master is receiving data from the slave or transmitting data to the slave. When an address is sent, each device in the system compares the first seven bits after a start condition with its address. If they match, the device considers itself addressed by the master.

I2C Sensor Interface Addresses and Timing

For linear acceleration the default (factory) 7-bit slave address is 0011001b. pp. 21

Table 14. SAD+Read/Write patterns

Command	SAD[7:1]	R/W	SAD+R/W
Read	0011001	1	00110011 (33h)
Write	0011001	0	00110010 (32h)

Table 15. Master receiving (reading) multiple bytes of data from slave

Master	ST	SAD +W		SUB		SR	SAD +R			MAK		MAK		NMAK	SP
Slave			SAK		SAK			SAK	DATA		DATA		DATA		

MAK is master acknowledge and NMAK is no master acknowledge

Register Mapping

Table 17. Register address map, pp 23

Name	Slave address	Type	Register address	
			Hex	Binary
Reserved (do not modify)	*Table 14*		00 - 1F	--
CTRL_REG1_A	*Table 14*	rw	20	010 0000
CTRL_REG2_A	*Table 14*	rw	21	010 0001
CTRL_REG3_A	*Table 14*	rw	22	010 0010
CTRL_REG4_A	*Table 14*	rw	23	010 0011
CTRL_REG5_A	*Table 14*	rw	24	010 0100
CTRL_REG6_A	*Table 14*	rw	25	010 0101
REFERENCE_A	*Table 14*	rw	26	010 0110

Control Register Description
Linear Acceleration Sensor

Table 18. CTRL_REG1_A register

CTRL_REG1_A address (20h)

ODR3	ODR2	ODR1	ODR0	LPen	Zen	Yen	Xen

Data rate selection

X, Y, and Z axis enabled

Example: enable x, y and z, no low power mode, and sample rate is 50 Hz, so we have

0 1 0 0 0 1 1 1

So, we have to write to address 0x20h this binary pattern for init and config, e.g.,

0 x 47

Data Register Description
Linear Acceleration Sensor

7.1.9
OUT_X_L_A (28h), OUT_X_H_A (29h)

X-axis acceleration data. The value is expressed in two's complement.

7.1.10
OUT_Y_L_A (2Ah), OUT_Y_H_A (2Bh)

Y-axis acceleration data. The value is expressed in two's complement.

7.1.11
OUT_Z_L_A (2Ch), OUT_Z_H_A (2Dh)

Z-axis acceleration data. The value is expressed in two's complement.

INT Register Description
Linear Acceleration Sensor

Table 40. INT1_CFG_A register — INT1_CFG_A (30h)

AOI	6D	ZHIE/ ZUPE	ZLIE/ ZDOWNE	YHIE/ YUPE	YLIE/ YDOWNE	XHIE/ XUPE	XLIE/ XDOWNE

Table 43. INT1_SRC_A register — INT1_SRC_A (31h)

0(1)	IA	ZH	ZL	YH	YL	XH	XL

Status register

The first register for configuration and the next register provides status of the interrupt.

Table 45. INT1_THS_A register — INT1_THS_A (32h)

0(1)	THS6	THS5	THS4	THS3	THS2	THS1	THS0

Threshold register

Steps for I2C Sensor Interface

Step 1. Identify the SAD (Slave address), e.g., the address of the sensor;

Step 2. Identify the control registers for init & config, then identify its/their addresses;

Step 3. Find init & config binary pattern to send to control registers for init & config;

Step 4. Identify the data registers and their addresses, then perform reading operation from these addresses (from these data registers);

Optional (Step 5). Identify INT configuration registers and their addresses, then find binary pattern for proper configuration of these INT registers;

Optional (Step 6). Identify INT data registers and their address, after proper init & config of INT control registers, read from these data registers;

Optional (Step 7) Idenfigy INT threshold registers and their address, set their threshold value to allow trigger INT once data beyond or below the threshold.