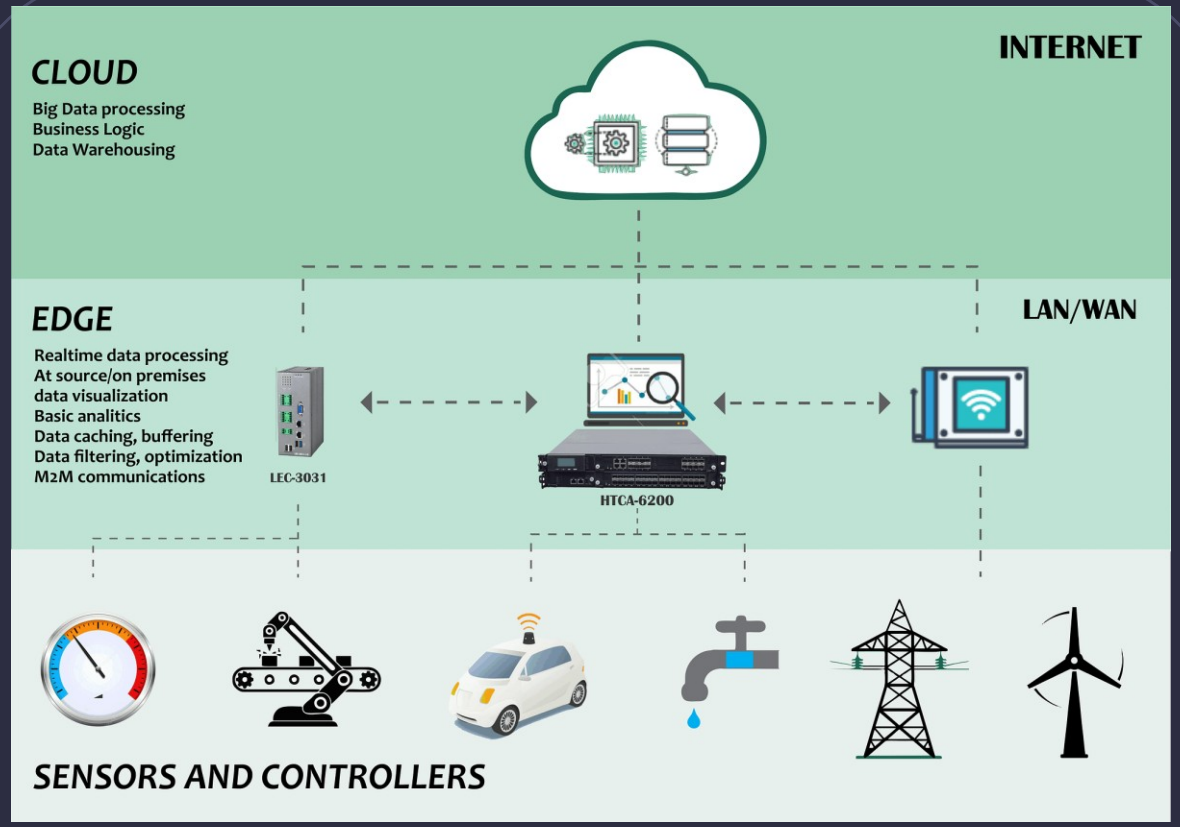
Internet of Things

Week 5: System Design with Sensors

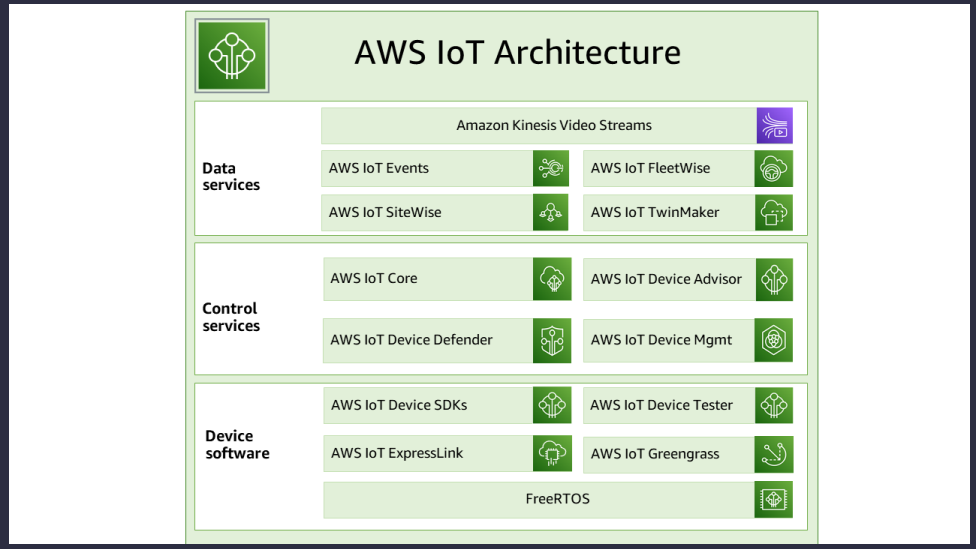
Fenerbahce University

How IoT Systems Work (End-to-End)

A practical mental model: device → network → edge/gateway → cloud → dashboard → actions



Edge–Fog–Cloud reference diagram



Example: AWS IoT “how it works” overview

System Design with Sensors I - Baremetal

How to Build an Embedded System?

- Buttons
- Sensors
  - Temperature
  - IMU
  - GPS
  - …
- Communication Interfaces
  - UART
  - SPI
  - I2C
  - Ethernet
  - …
- RF Transceiver

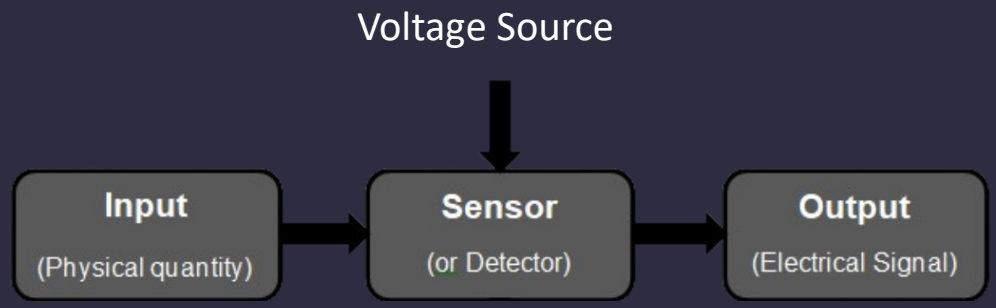
- LEDs
- Motors
- Communication Interfaces
  - UART
  - SPI
  - I2C
  - Ethernet
  - …
- RF Transceiver

System Design with Sensors I - Baremetal

Sensors

Sensor

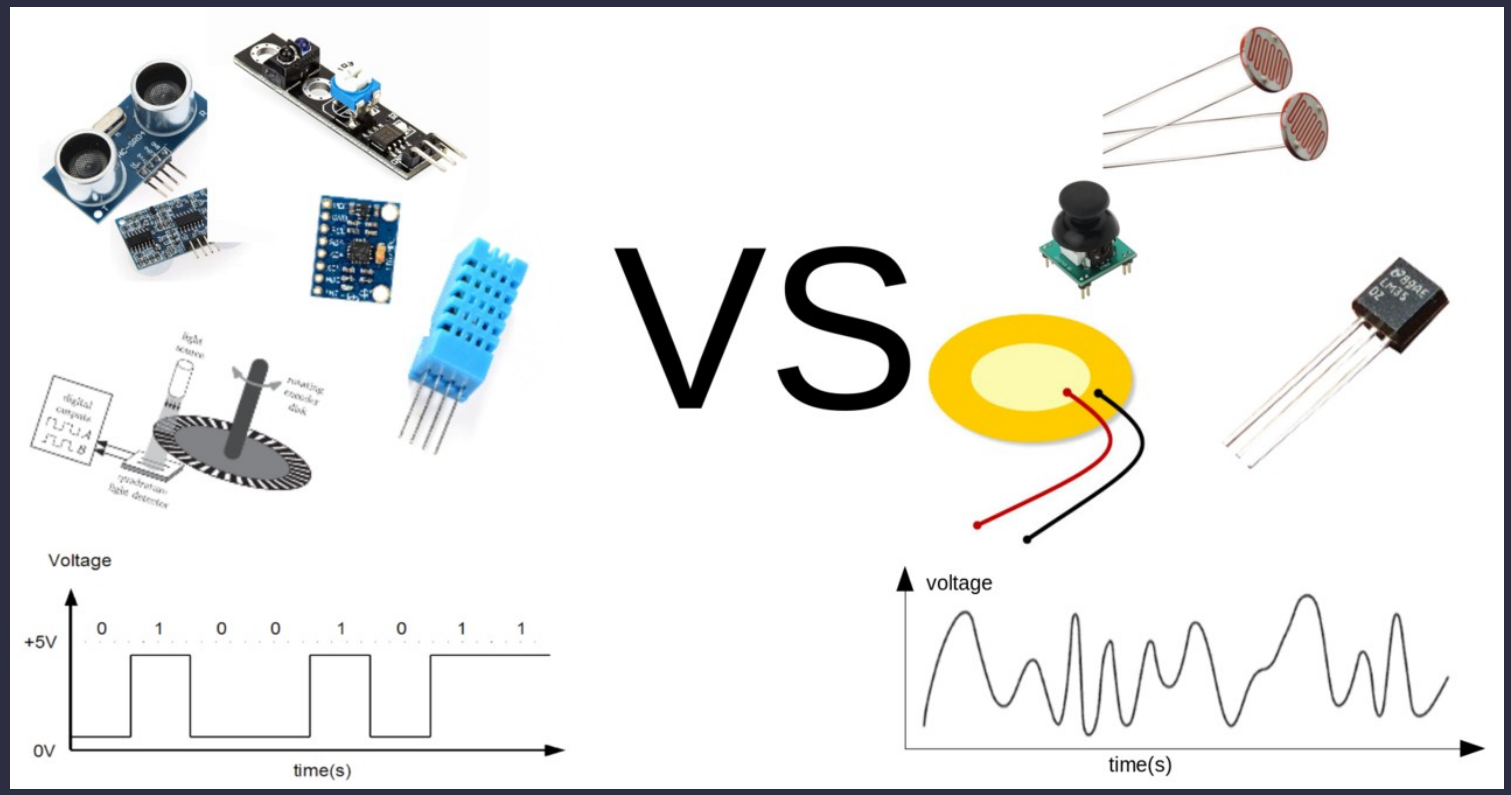
- Device for Detection: A sensor is a device designed to detect changes in physical, chemical, or biological environments.
- Signal Conversion: It converts these changes into electrical signals that can be measured and processed.
- Real-World Interface: Serves as the bridge between the physical world and embedded systems.
- Core Functionality: Enables systems to monitor, control, and respond to environmental variations.



System Design with Sensors I - Baremetal

Sensors

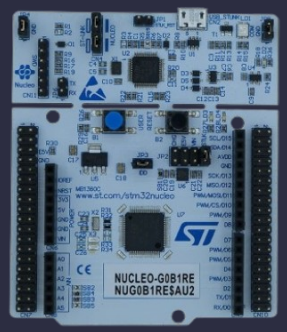
- Sensor Output Types
  - Analog Sensors
  - Digital Sensors
- Sensors Types
  - Temperature
  - Pressure
  - Light
  - Motion
  - Gas
  - GPS
  - Etc…



System Design with Sensors I - Baremetal

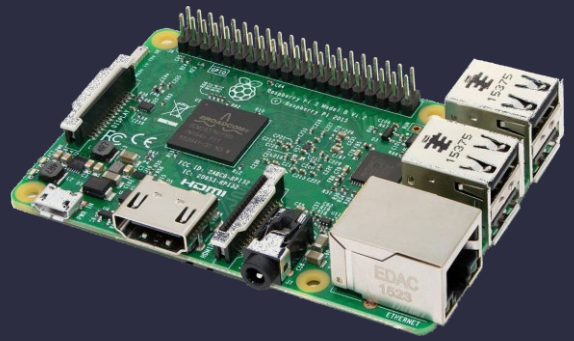
- MCU & MPU Development Boards

Arduino Mega



STM32 Nucleo 64

MCU Based



Rasberry Pi 3

MPU Based

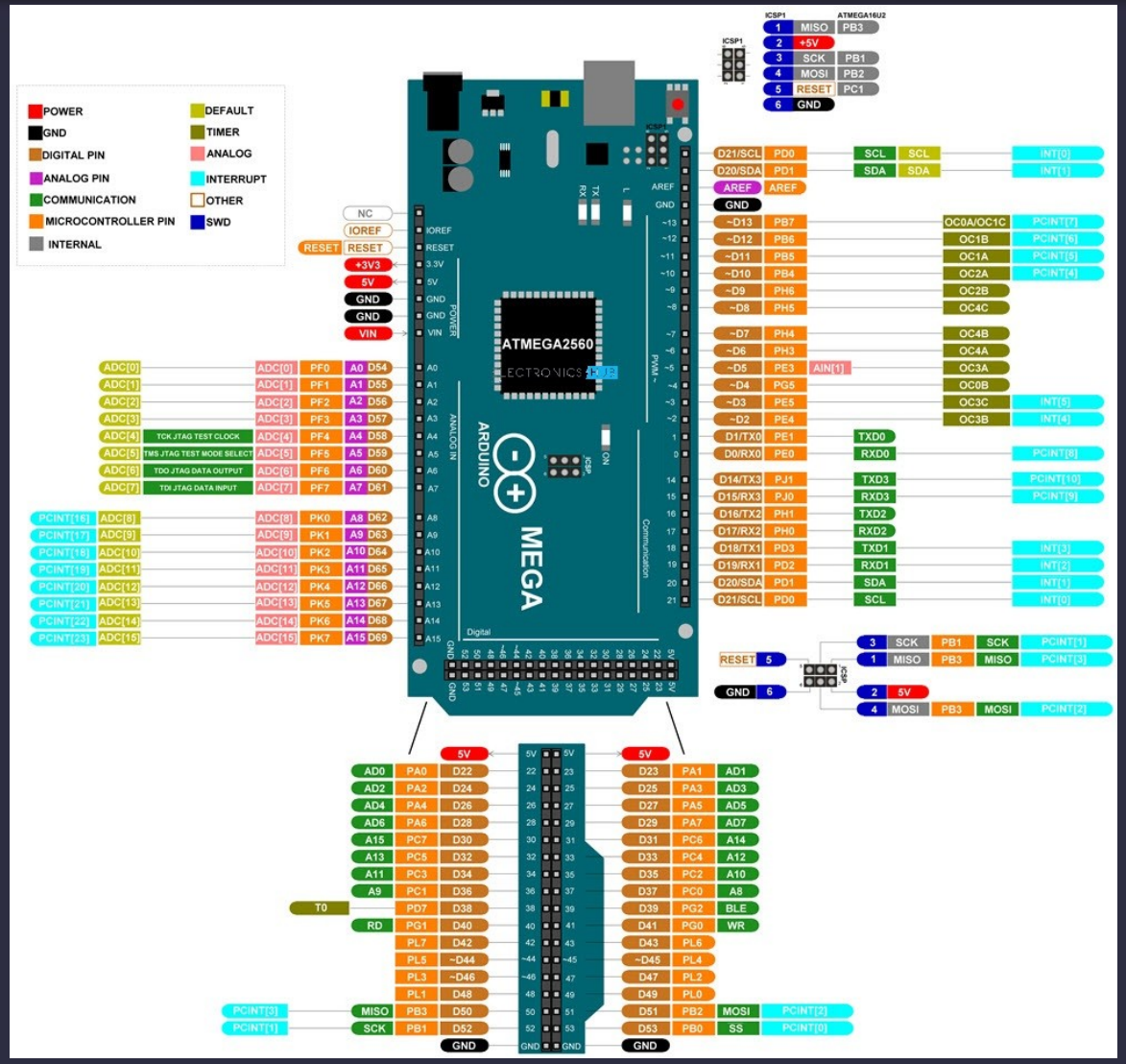
System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - Arduino Mega 2560 Development Board + Extension Card
  - The board accommodates
    - The ATmega2560 microcontroller, which operates at a frequency of 16 MHz. 54 digital input/output pins,
    - 16 analog inputs,
    - 4 UARTs
    - A USB connection,
    - A power jack
    - An ICSP header
    - A reset button.
  - For Details:

https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf

System Design with Sensors I - Baremetal

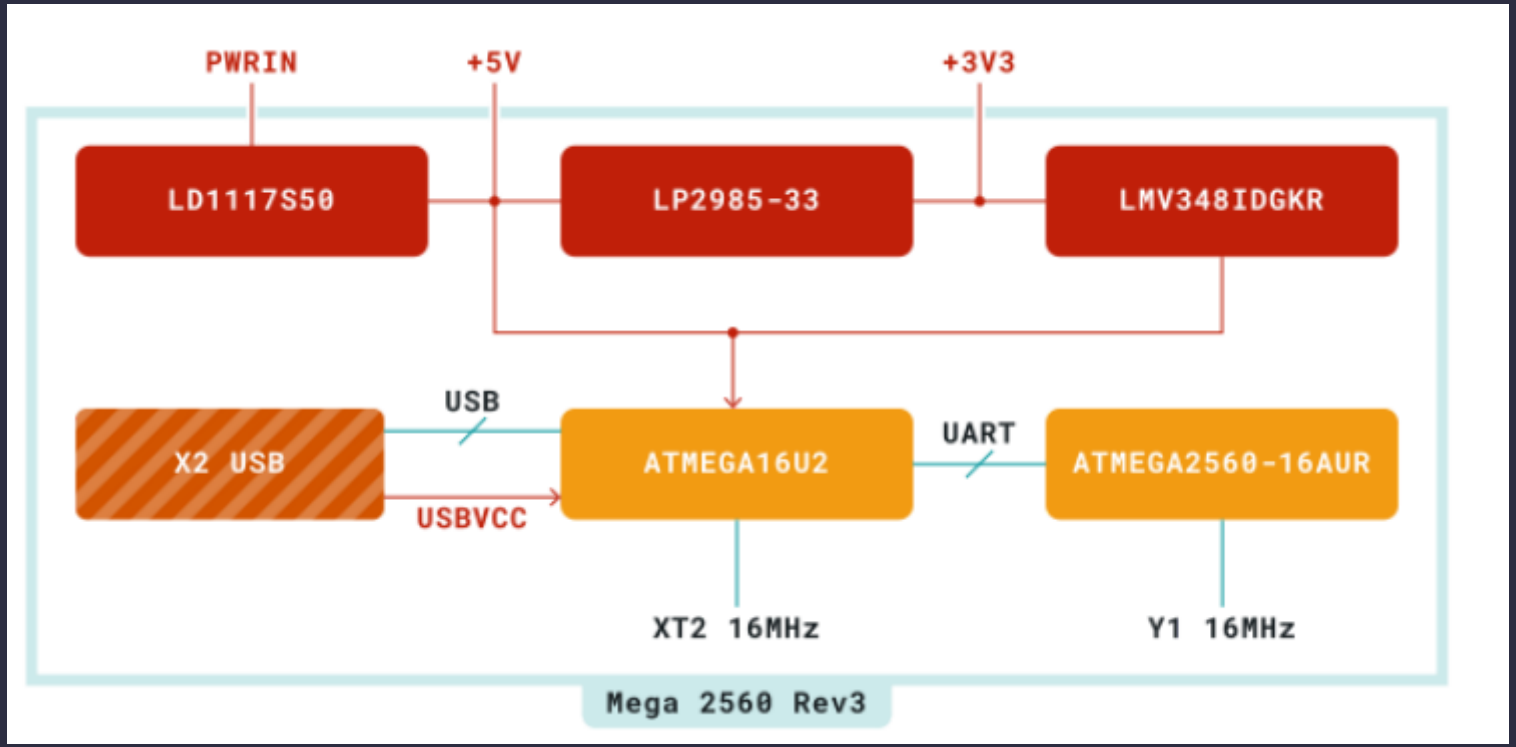
- MCU & MPU Development Boards
  - Arduino Mega 2560 Development Board + Extension Card
  - Board Pinout Diagram



System Design with Sensors I - Baremetal

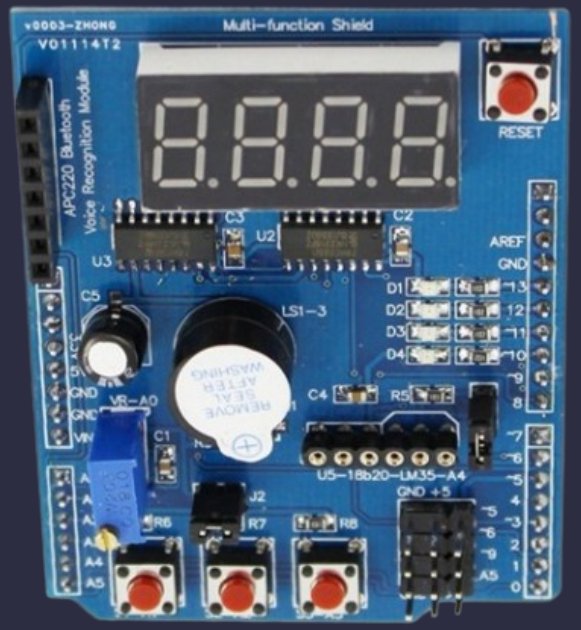
- MCU & MPU Development Boards
  - Arduino Mega 2560 Development Board + Extension Card

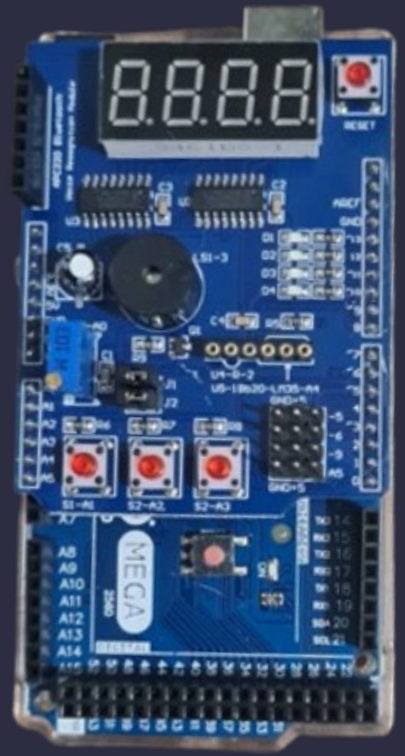
- Board Block Diagram

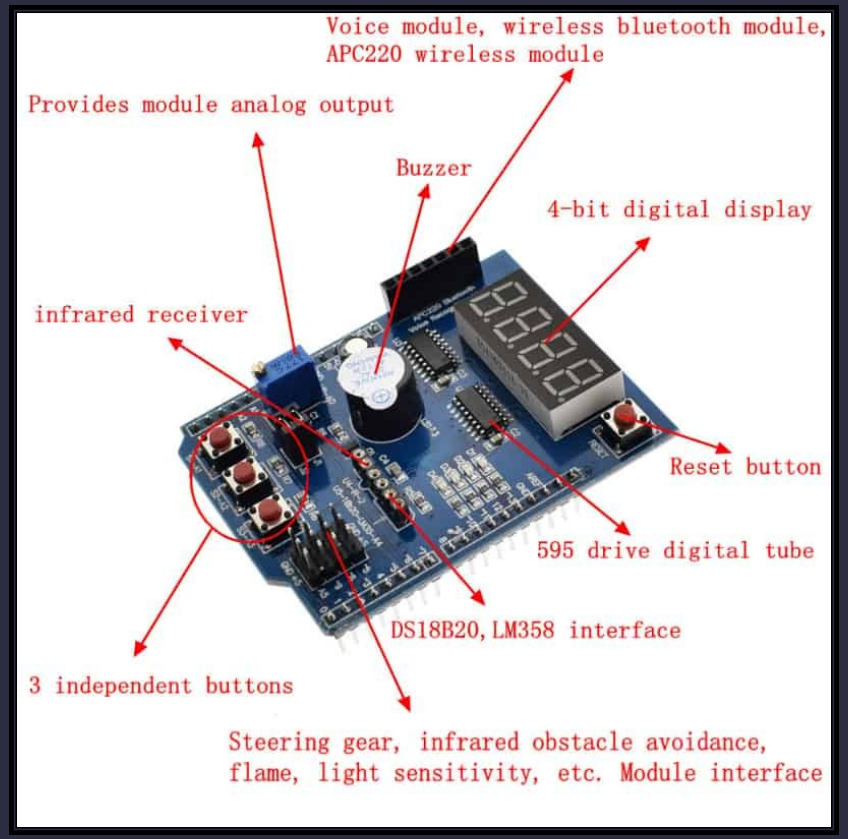


System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - Arduino Mega 2560 Development Board + Extension Card

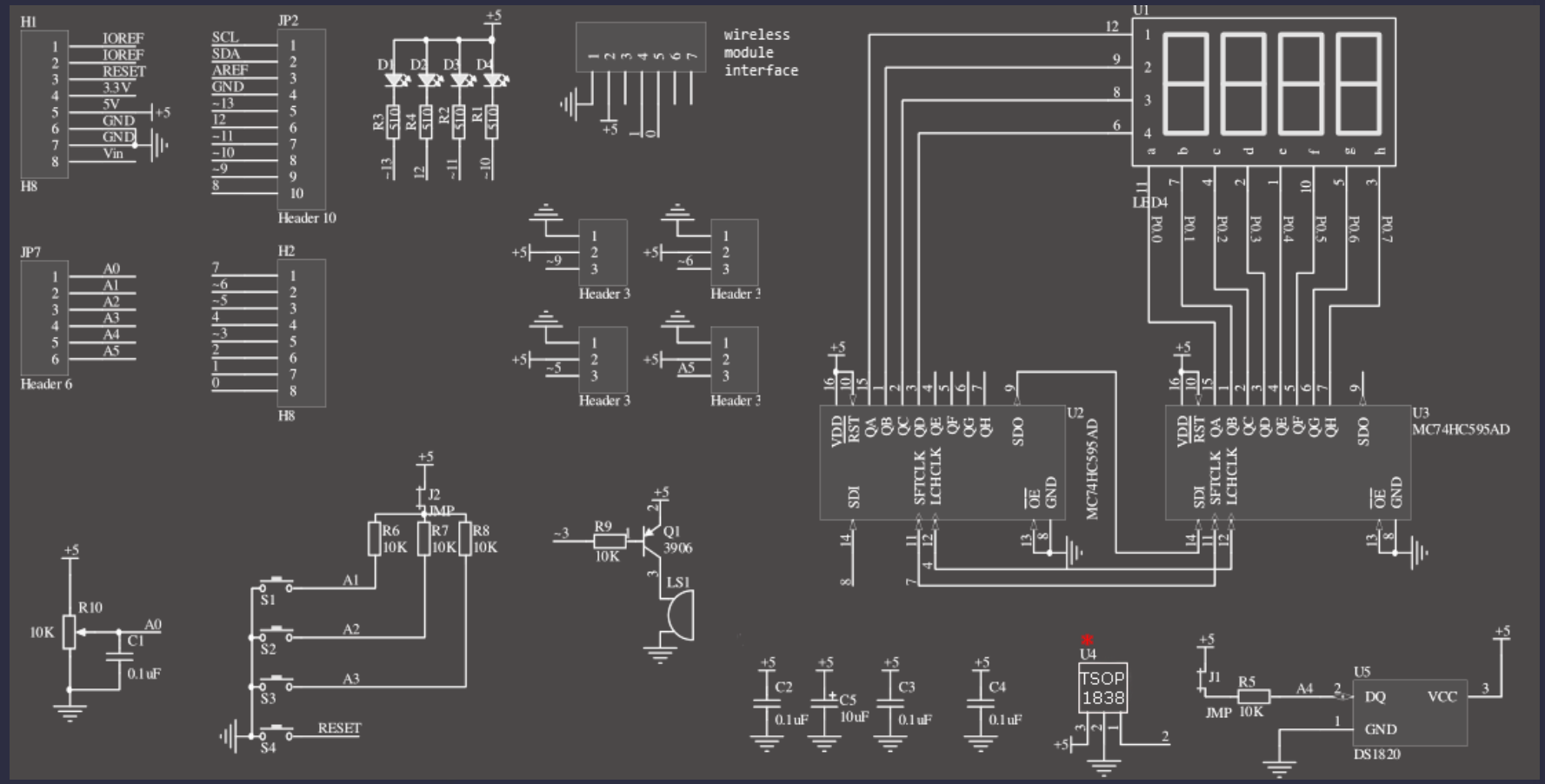






System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - Arduino Mega 2560 Development Board + Extension Card



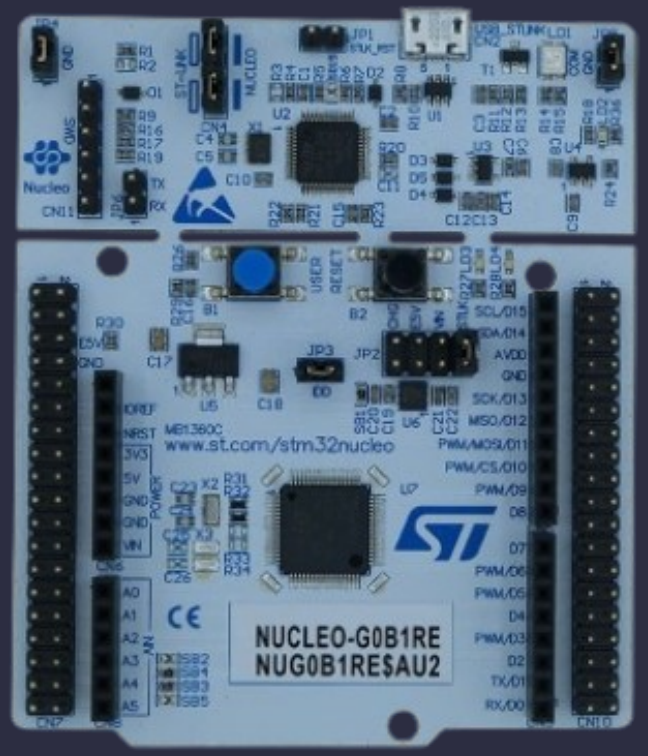
System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - STM32 Nucleo Development Board

  - STM32 microcontroller in LQFP64 package
  - Three LEDs:
    - USB communication (LD1), user LED (LD2), power LED (LD3)
  - Two push-buttons: USER and RESET
  - Two types of extension resources
    - ARDUINO® Uno V3 connectivity
  - ST morpho extension pin headers for full access to all STM32 I/Os
  - Three different interfaces supported on USB:
    - Virtual COM port
    - Mass storage
    - Debug port

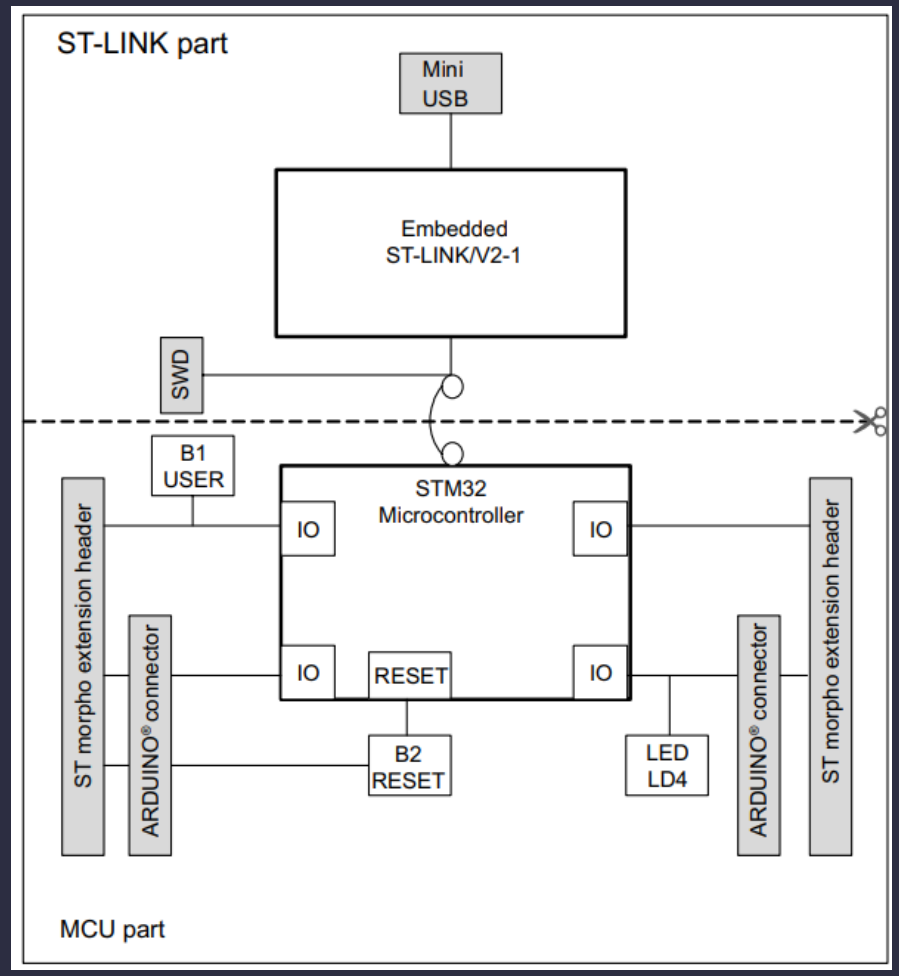
  - For details:

  https://www.st.com/resource/en/user_manual/um1724-stm32-nucleo64-boards-mb1136-stmicroelectronics.pdf



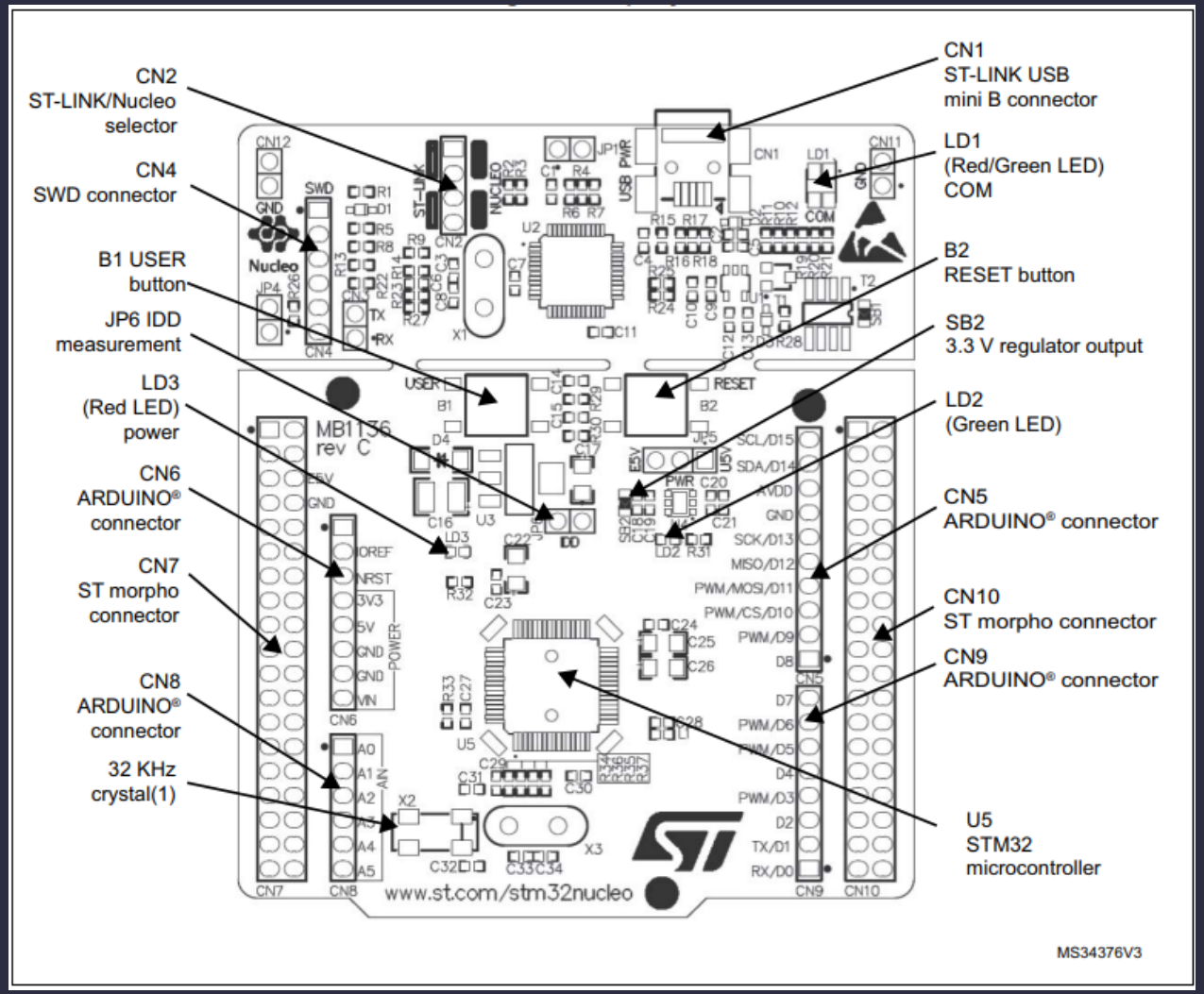
System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - STM32 Nucleo Development Board
  - STM32 Nucleo Board Block Diagram



System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - STM32 Nucleo Development Board
  - STM32 Nucleo Board Block Diagram



MCU & MPU Architectures, Interfaces

- MCU & MPU Development Boards

- MPU, STM32 Nucleo with Extension Board

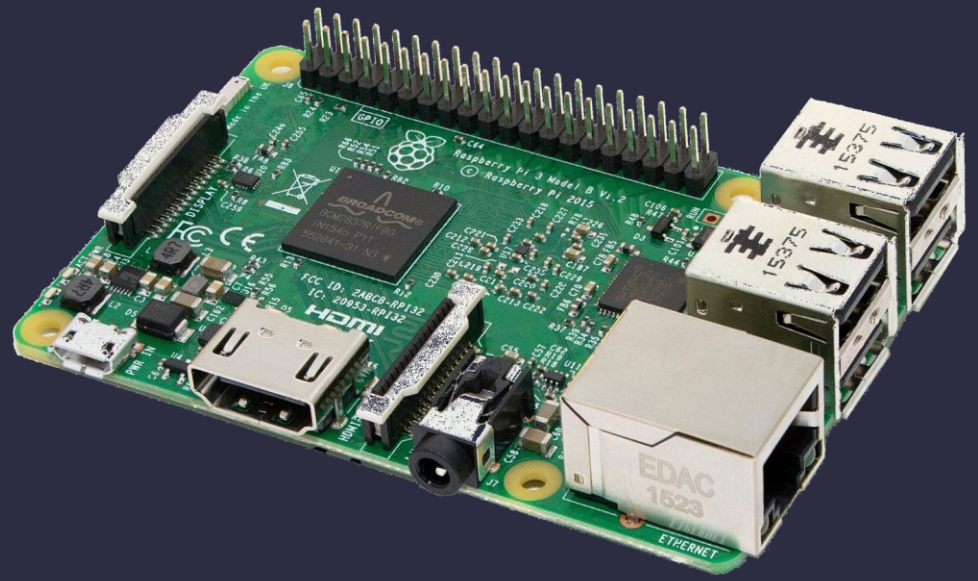
System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - Rasberry PI 3 Development Board

- Raspberry Pi 3 Model B+ has a 64-bit quad-core processor running at 1.4GHz
- Dual-band 2.4GHz and 5GHz wireless LAN
- Bluetooth 4.2/BLE
- Gigabit Ethernet
- USB 2.0
- 1 x full size HDMI
- MIPI DSI display port
- MIPI CSI camera port

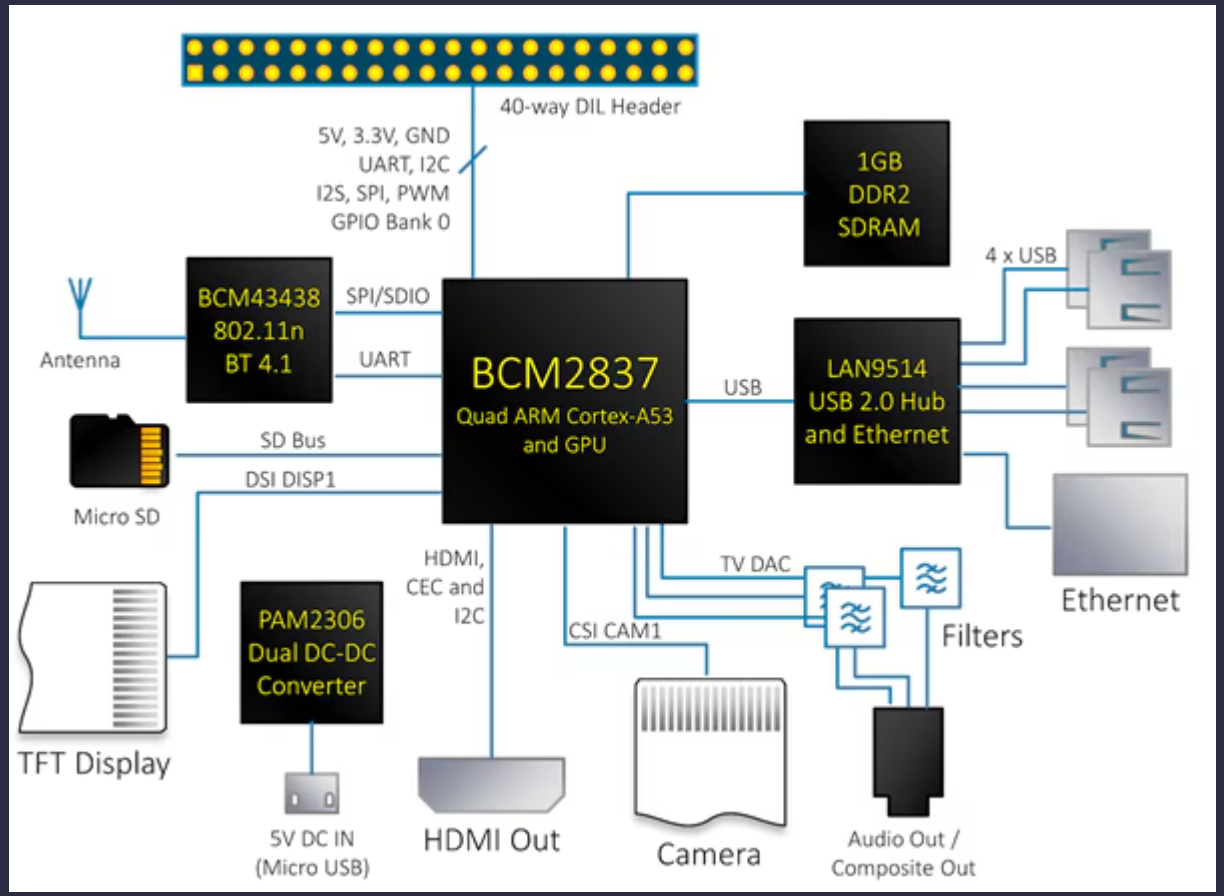
- Details:

https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf



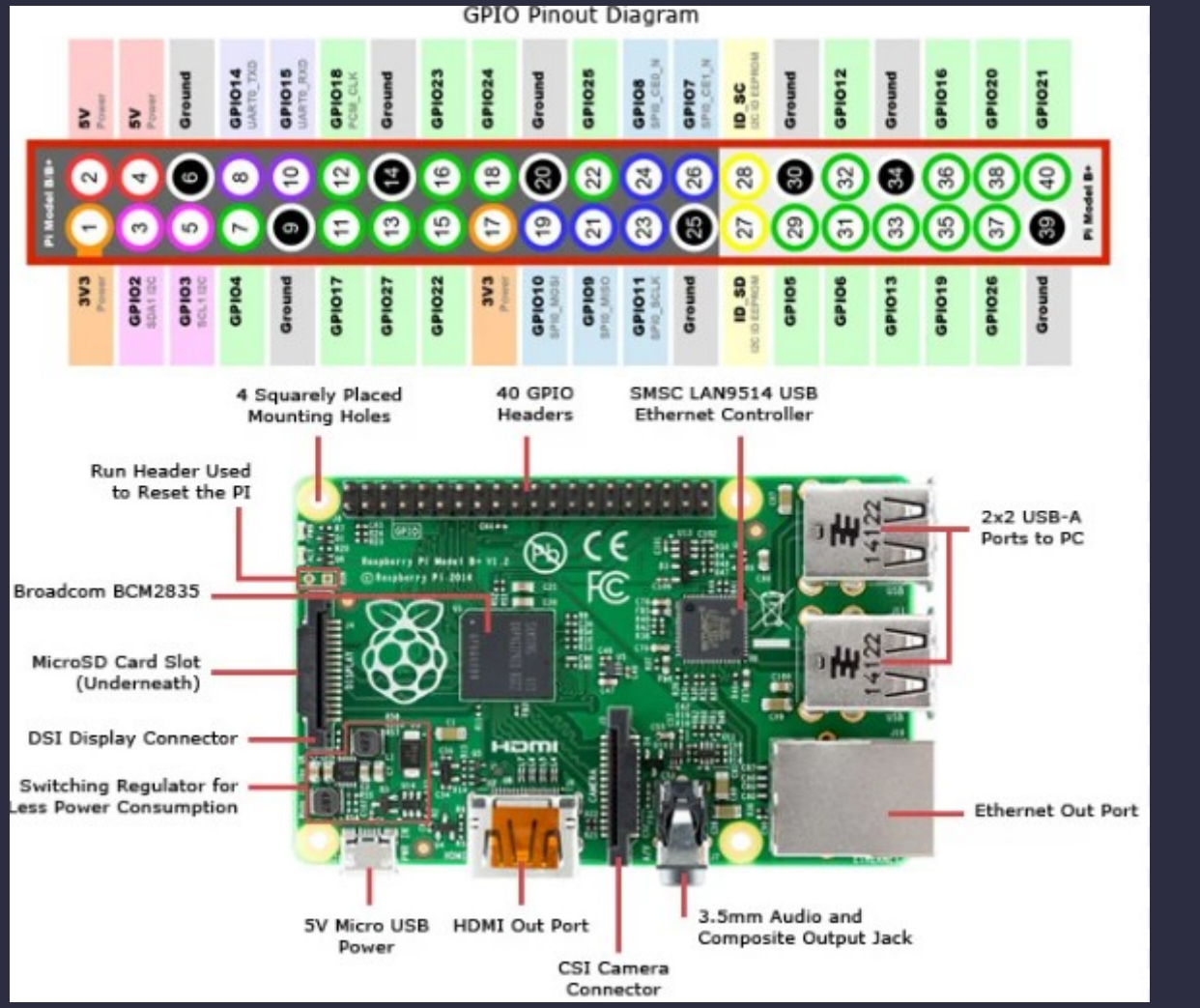
System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - Rasberry PI 3 Development Board
  - Block Diagram



System Design with Sensors I - Baremetal

- MCU & MPU Development Boards
  - Rasberry PI 3 Development Board
  - Block Diagram

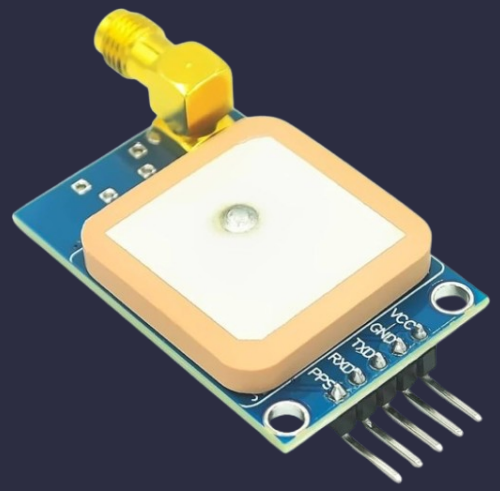


System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors

  - Use Case Hardwares

  - Gyroscope, MPU6050
  - Laser Distance Measurement, TOF050C
  - GPS, NEO-7M
  - RF Transceiver, Dorji DRF1278DM

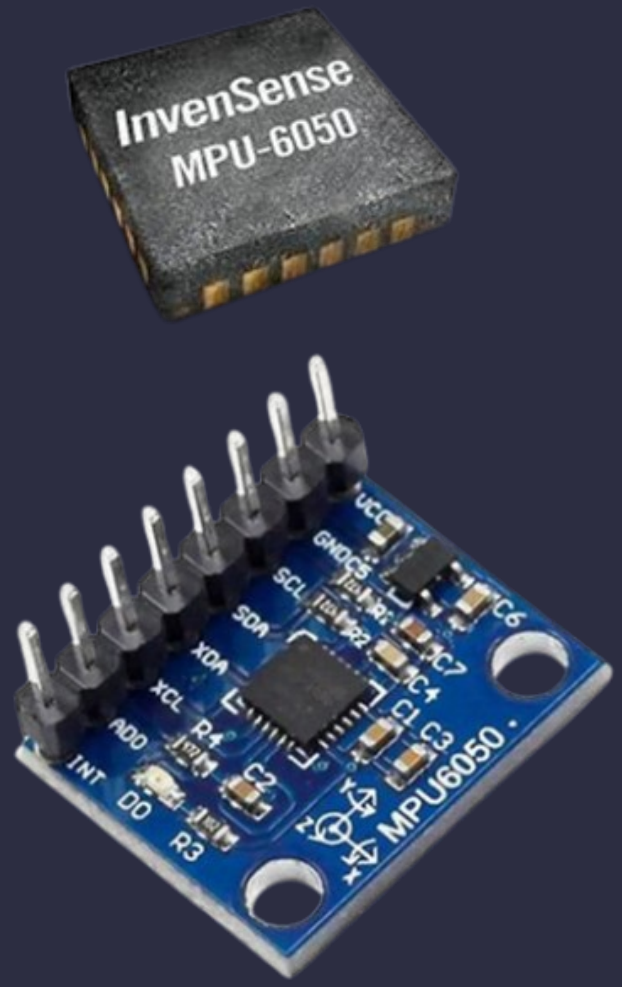


System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050
  - Voltage: 3-5V
  - Gyro Maximum Degree Measurement: +- 250, 500, 1000, 2000 ° / s
  - Accelerometer: ± 2 ± 4 ± 8 ± 16 g
  - Communication: Standart I²C
  - I2C Address: 0x68
  - Up to 400KHz I2C Speed
  - Details

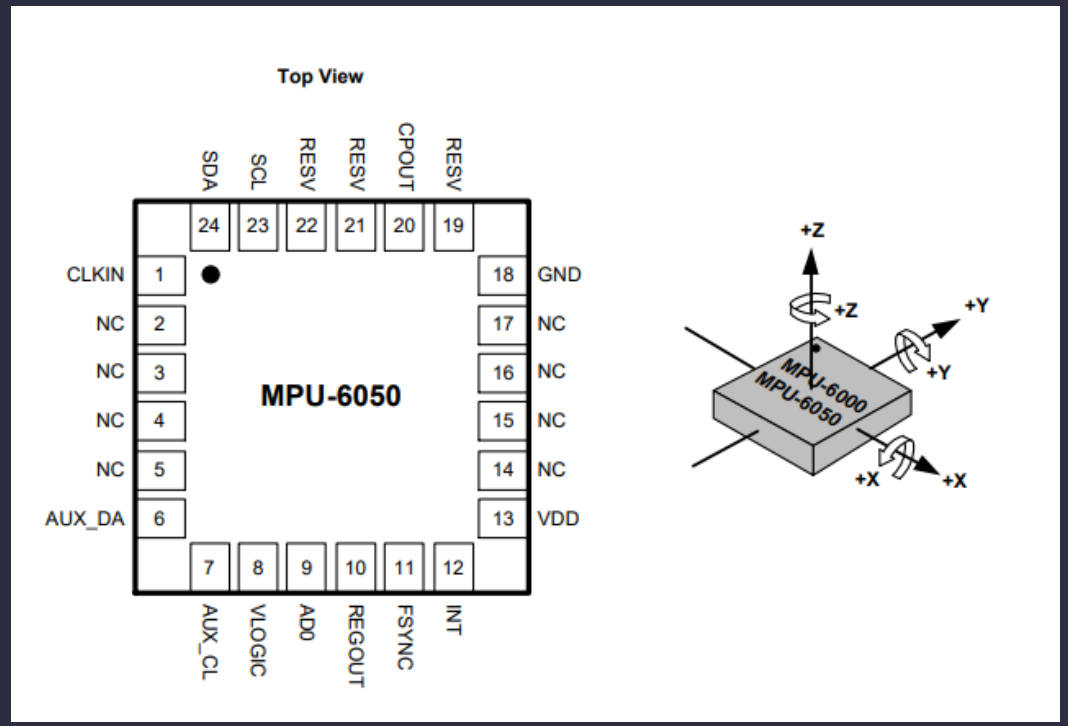
https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

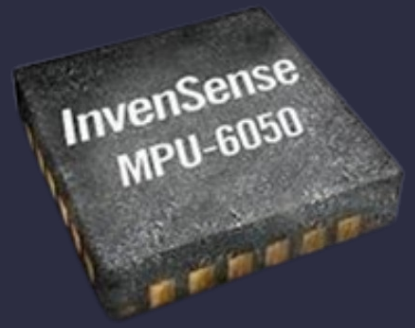
https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf



System Design with Sensors I - Baremetal

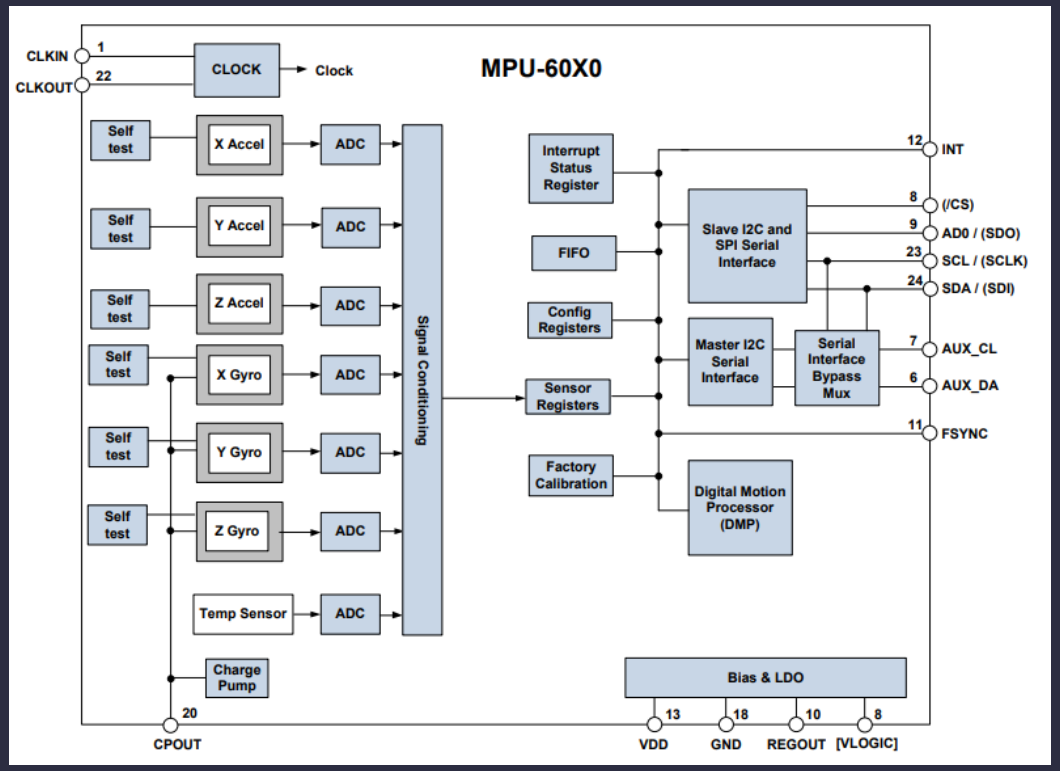
- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 I/Os

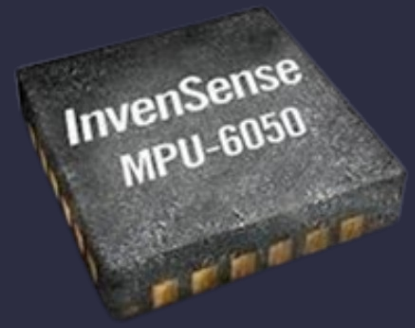




System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Diagram



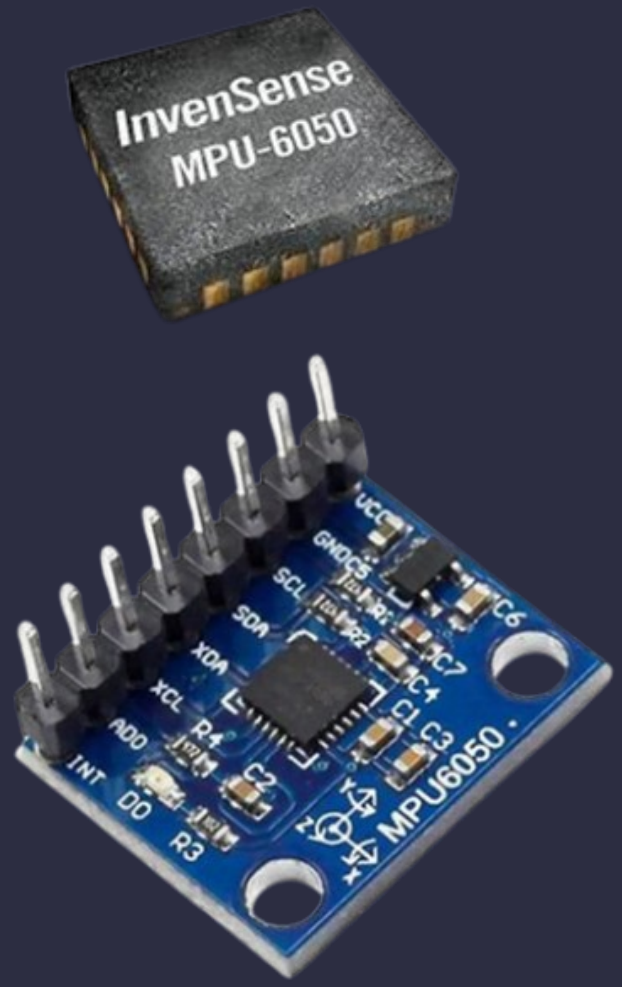


System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Diagram

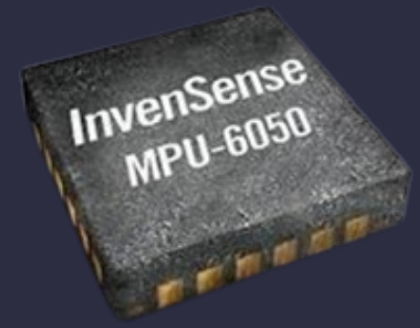
  Interrupt Sources

  - FIFO Overflow FIFO
  - Data Ready Sensor Registers
  - I2C Master errors: Lost Arbitration, NACKs I 2C Master



System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Diagram
  - Typical Register Read



Single-Byte Read Sequence

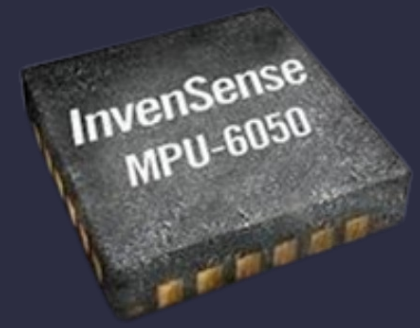
Master	S	AD+W		RA		S	AD+R			NACK	P
Slave			ACK		ACK			ACK	DATA		

Burst Read Sequence

Master	S	AD+W		RA		S	AD+R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Diagram
  - Typical Register Write



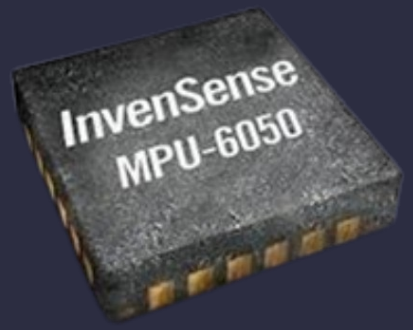
Master	S	AD+W		RA		DATA		P
Slave			ACK		ACK		ACK	

Burst Write Sequence

Master	S	AD+W		RA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	

System Design with Sensors I - Baremetal

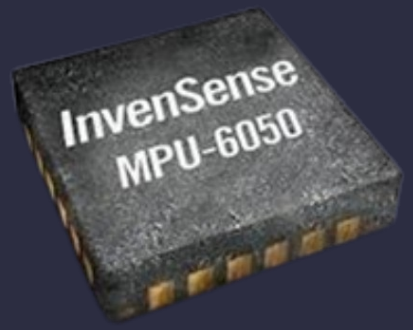
- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Register Space



Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0D	13	SELF_TEST_X	R/W	XA_TEST[4-2]			XG_TEST[4-0]				
0E	14	SELF_TEST_Y	R/W	YA_TEST[4-2]			YG_TEST[4-0]				
0F	15	SELF_TEST_Z	R/W	ZA_TEST[4-2]			ZG_TEST[4-0]				
10	16	SELF_TEST_A	R/W	RESERVED		XA_TEST[1-0]		YA_TEST[1-0]		ZA_TEST[1-0]	
19	25	SMPLRT_DIV	R/W	SMPLRT_DIV[7:0]							
1A	26	CONFIG	R/W	-	-	EXT_SYNC_SET[2:0]			DLPF_CFG[2:0]		
1B	27	GYRO_CONFIG	R/W	-	-	-	FS_SEL [1:0]		-	-	-
1C	28	ACCEL_CONFIG	R/W	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]				
23	35	FIFO_EN	R/W	TEMP _FIFO_EN	XG _FIFO_EN	YG _FIFO_EN	ZG _FIFO_EN	ACCEL _FIFO_EN	SLV2 _FIFO_EN	SLV1 _FIFO_EN	SLV0 _FIFO_EN
24	36	I2C_MST_CTRL	R/W	MULT _MST_EN	WAIT _FOR_ES	SLV_3 _FIFO_EN	I2C_MST _P_NSR	I2C_MST_CLK[3:0]			
25	37	I2C_SLV0_ADDR	R/W	I2C_SLV0 _RW	I2C_SLV0_ADDR[6:0]						
26	38	I2C_SLV0_REG	R/W	I2C_SLV0_REG[7:0]							
27	39	I2C_SLV0_CTRL	R/W	I2C_SLV0 _EN	I2C_SLV0 _BYTE_SW	I2C_SLV0 _REG_DIS	I2C_SLV0 _GRP	I2C_SLV0_LEN[3:0]			
28	40	I2C_SLV1_ADDR	R/W	I2C_SLV1 _RW	I2C_SLV1_ADDR[6:0]						
29	41	I2C_SLV1_REG	R/W	I2C_SLV1_REG[7:0]							
2A	42	I2C_SLV1_CTRL	R/W	I2C_SLV1 _EN	I2C_SLV1 _BYTE_SW	I2C_SLV1 _REG_DIS	I2C_SLV1 _GRP	I2C_SLV1_LEN[3:0]			
2B	43	I2C_SLV2_ADDR	R/W	I2C_SLV2 _RW	I2C_SLV2_ADDR[6:0]						
2C	44	I2C_SLV2_REG	R/W	I2C_SLV2_REG[7:0]							
2D	45	I2C_SLV2_CTRL	R/W	I2C_SLV2 _EN	I2C_SLV2 _BYTE_SW	I2C_SLV2 _REG_DIS	I2C_SLV2 _GRP	I2C_SLV2_LEN[3:0]			
2E	46	I2C_SLV3_ADDR	R/W	I2C_SLV3 _RW	I2C_SLV3_ADDR[6:0]						
2F	47	I2C_SLV3_REG	R/W	I2C_SLV3_REG[7:0]							
30	48	I2C_SLV3_CTRL	R/W	I2C_SLV3 _EN	I2C_SLV3 _BYTE_SW	I2C_SLV3 _REG_DIS	I2C_SLV3 _GRP	I2C_SLV3_LEN[3:0]			
31	49	I2C_SLV4_ADDR	R/W	I2C_SLV4 _RW	I2C_SLV4_ADDR[6:0]						
32	50	I2C_SLV4_REG	R/W	I2C_SLV4_REG[7:0]							
33	51	I2C_SLV4_DO	R/W	I2C_SLV4_DO[7:0]							
34	52	I2C_SLV4_CTRL	R/W	I2C_SLV4 _EN	I2C_SLV4 _INT_EN	I2C_SLV4 _REG_DIS	I2C_MST_DLY[4:0]				
35	53	I2C_SLV4_DI	R	I2C_SLV4_DI[7:0]							
36	54	I2C_MST_STATUS	R	PASS_ THROUGH	I2C_SLV4 _DONE	I2C_LOST _ARB	I2C_SLV4 _NACK	I2C_SLV3 _NACK	I2C_SLV2 _NACK	I2C_SLV1 _NACK	I2C_SLV0 _NACK
37	55	INT_PIN_CFG	R/W	INT_LEVEL	INT_OPEN	LATCH _INT_EN	INT_RD _CLEAR	FSYNC_ INT_LEVEL	FSYNC _INT_EN	I2C _BYPASS _EN	-
38	56	INT_ENABLE	R/W	-	-	-	FIFO _OFLOW _EN	I2C_MST _INT_EN	-	-	DATA _RDY_EN
3A	58	INT_STATUS	R	-	-	-	FIFO _OFLOW _INT	I2C_MST _INT	-	-	DATA _RDY_INT

System Design with Sensors I - Baremetal

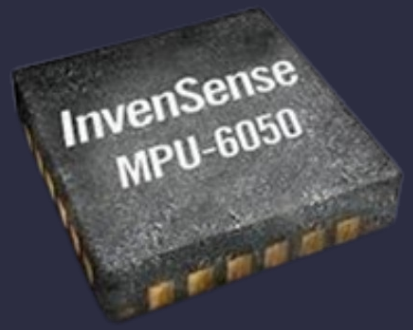
- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Register Space



Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT_H	R	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT_L	R	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT_H	R	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT_L	R	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT_H	R	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT_L	R	ACCEL_ZOUT[7:0]							
41	65	TEMP_OUT_H	R	TEMP_OUT[15:8]							
42	66	TEMP_OUT_L	R	TEMP_OUT[7:0]							
43	67	GYRO_XOUT_H	R	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT_L	R	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT_H	R	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT_L	R	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT_H	R	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT_L	R	GYRO_ZOUT[7:0]							
49	73	EXT_SENS_DATA_00	R	EXT_SENS_DATA_00[7:0]							
4A	74	EXT_SENS_DATA_01	R	EXT_SENS_DATA_01[7:0]							
4B	75	EXT_SENS_DATA_02	R	EXT_SENS_DATA_02[7:0]							
4C	76	EXT_SENS_DATA_03	R	EXT_SENS_DATA_03[7:0]							
4D	77	EXT_SENS_DATA_04	R	EXT_SENS_DATA_04[7:0]							
4E	78	EXT_SENS_DATA_05	R	EXT_SENS_DATA_05[7:0]							
4F	79	EXT_SENS_DATA_06	R	EXT_SENS_DATA_06[7:0]							
50	80	EXT_SENS_DATA_07	R	EXT_SENS_DATA_07[7:0]							
51	81	EXT_SENS_DATA_08	R	EXT_SENS_DATA_08[7:0]							
52	82	EXT_SENS_DATA_09	R	EXT_SENS_DATA_09[7:0]							
53	83	EXT_SENS_DATA_10	R	EXT_SENS_DATA_10[7:0]							
54	84	EXT_SENS_DATA_11	R	EXT_SENS_DATA_11[7:0]							
55	85	EXT_SENS_DATA_12	R	EXT_SENS_DATA_12[7:0]							
56	86	EXT_SENS_DATA_13	R	EXT_SENS_DATA_13[7:0]							
57	87	EXT_SENS_DATA_14	R	EXT_SENS_DATA_14[7:0]							
58	88	EXT_SENS_DATA_15	R	EXT_SENS_DATA_15[7:0]							
59	89	EXT_SENS_DATA_16	R	EXT_SENS_DATA_16[7:0]							
5A	90	EXT_SENS_DATA_17	R	EXT_SENS_DATA_17[7:0]							
5B	91	EXT_SENS_DATA_18	R	EXT_SENS_DATA_18[7:0]							
5C	92	EXT_SENS_DATA_19	R	EXT_SENS_DATA_19[7:0]							
5D	93	EXT_SENS_DATA_20	R	EXT_SENS_DATA_20[7:0]							
5E	94	EXT_SENS_DATA_21	R	EXT_SENS_DATA_21[7:0]							
5F	95	EXT_SENS_DATA_22	R	EXT_SENS_DATA_22[7:0]							
60	96	EXT_SENS_DATA_23	R	EXT_SENS_DATA_23[7:0]							
63	99	I2C_SLV0_DO	R/W	I2C_SLV0_DO[7:0]							
64	100	I2C_SLV1_DO	R/W	I2C_SLV1_DO[7:0]							
65	101	I2C_SLV2_DO	R/W	I2C_SLV2_DO[7:0]							
66	102	I2C_SLV3_DO	R/W	I2C_SLV3_DO[7:0]							

System Design with Sensors I - Baremetal

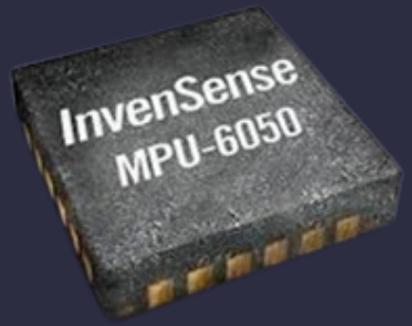
- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Register Space



Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
67	103	I2C_MST_DELAY_CTRL	R/W	DELAY_ES_SHADOW	-	-	I2C_SLV4_DLY_EN	I2C_SLV3_DLY_EN	I2C_SLV2_DLY_EN	I2C_SLV1_DLY_EN	I2C_SLV0_DLY_EN
68	104	SIGNAL_PATH_RESET	R/W	-	-	-	-	-	GYRO_RESET	ACCEL_RESET	TEMP_RESET
6A	106	USER_CTRL	R/W	-	FIFO_EN	I2C_MST_EN	I2C_IF_DIS	-	FIFO_RESET	I2C_MST_RESET	SIG_COND_RESET
6B	107	PWR_MGMT_1	R/W	DEVICE_RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		
6C	108	PWR_MGMT_2	R/W	LP_WAKE_CTRL[1:0]		STBY_XA	STBY_YA	STBY_ZA	STBY_XG	STBY_YG	STBY_ZG
72	114	FIFO_COUNTH	R/W	FIFO_COUNT[15:8]							
73	115	FIFO_COUNTL	R/W	FIFO_COUNT[7:0]							
74	116	FIFO_R_W	R/W	FIFO_DATA[7:0]							
75	117	WHO_AM_I	R	-	WHO_AM_I[6:1]						-

System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Gyroscope, MPU6050 Register Space
  - Disable sleep mode
  - Read Registers 59 to 72 for Gyro and Acc Data



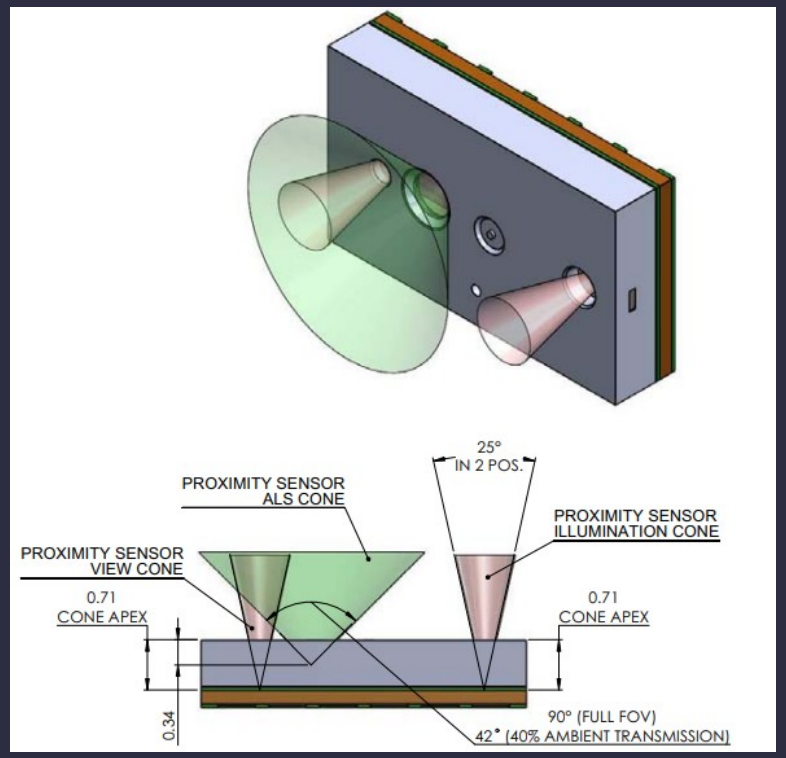
System Design with Sensors I - Baremetal

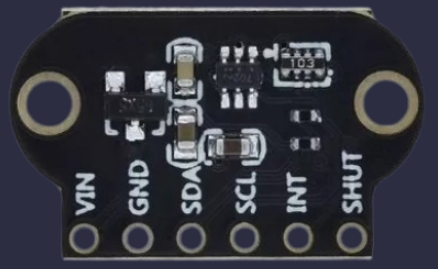
- Sensor Development Boards
  - Sensors
  - Laser Distance Measurement, TOF050C
  - ToF distance sensor module based on the VL6180X system. It uses FlightSense™ technology that allows you to measure the absolute distance regardless of the color and surface of the detected object.
  - Range: 2 to 50 cm
  - Dead zone: 0 to 2 cm
  - Interface: I2C
  - I2C address: 0x29
  - Viewing Angle (FOV): 25°
  - Power supply: 3 to 5 V

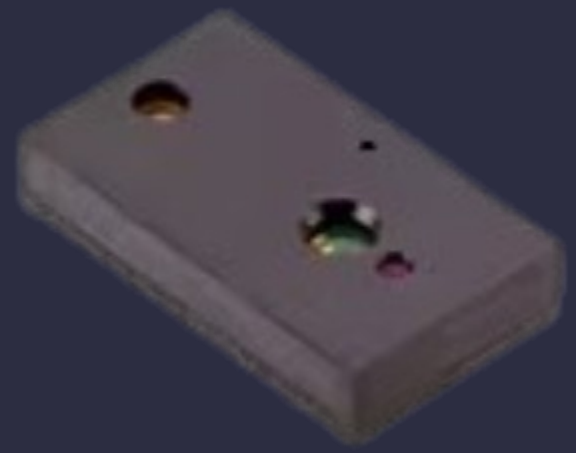
Details:
https://www.st.com/en/imaging-and-photonics-solutions/vl6180x.html

System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Laser Distance Measurement, TOF050C

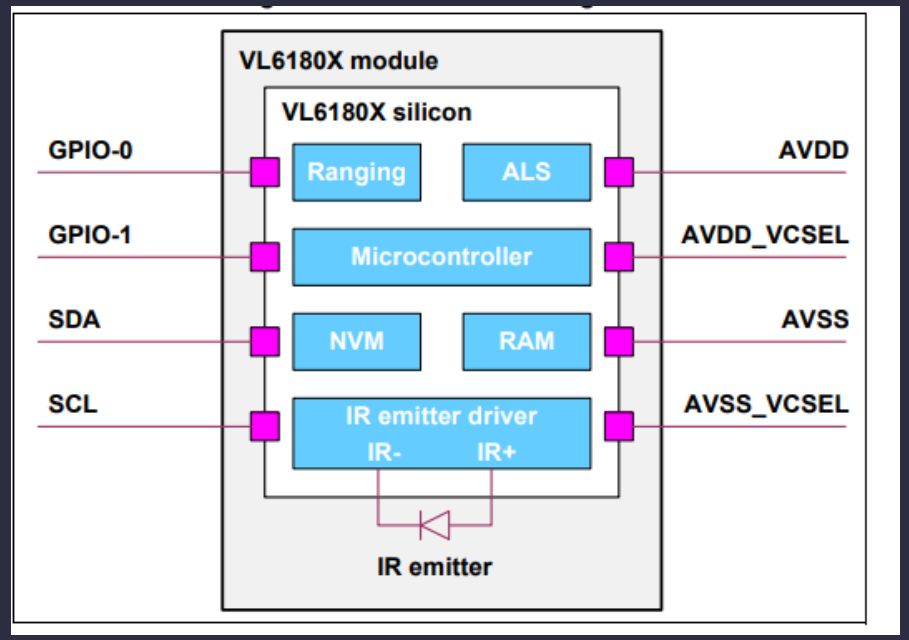


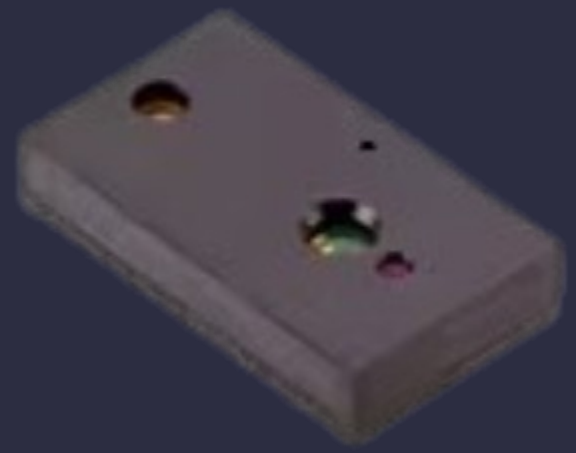




System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Laser Distance Measurement, TOF050C, I/Os





System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Laser Distance Measurement, TOF050C, Register Map

Offset	Register name
0x000	IDENTIFICATION__MODEL_ID
0x001	IDENTIFICATION__MODEL_REV_MAJOR
0x002	IDENTIFICATION__MODEL_REV_MINOR
0x003	IDENTIFICATION__MODULE_REV_MAJOR
0x004	IDENTIFICATION__MODULE_REV_MINOR
0x006	IDENTIFICATION__DATE_HI
0x007	IDENTIFICATION__DATE_LO
0x008:0x009	IDENTIFICATION__TIME
0x010	SYSTEM__MODE_GPIO0
0x011	SYSTEM__MODE_GPIO1
0x012	SYSTEM__HISTORY_CTRL
0x014	SYSTEM__INTERRUPT_CONFIG_GPIO
0x015	SYSTEM__INTERRUPT_CLEAR
0x016	SYSTEM__FRESH_OUT_OF_RESET
0x017	SYSTEM__GROUPED_PARAMETER_HOLD
0x018	SYSRANGE__START
0x019	SYSRANGE__THRESH_HIGH
0x01A	SYSRANGE__THRESH_LOW
0x01B	SYSRANGE__INTERMEASUREMENT_PERIOD
0x01C	SYSRANGE__MAX_CONVERGENCE_TIME
0x01E	SYSRANGE__CROSSTALK_COMPENSATION_RATE
0x021	SYSRANGE__CROSSTALK_VALID_HEIGHT
0x022	SYSRANGE__EARLY_CONVERGENCE_ESTIMATE
0x024	SYSRANGE__PART_TO_PART_RANGE_OFFSET
0x025	SYSRANGE__RANGE_IGNORE_VALID_HEIGHT
0x026	SYSRANGE__RANGE_IGNORE_THRESHOLD
0x02C	SYSRANGE__MAX_AMBIENT_LEVEL_MULT
0x02D	SYSRANGE__RANGE_CHECK_ENABLES
0x02E	SYSRANGE__VHV_RECALIBRATE
0x031	SYSRANGE__VHV_REPEAT_RATE
0x038	SYSALS__START
0x03A	SYSALS__THRESH_HIGH
0x03C	SYSALS__THRESH_LOW

System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Laser Distance Measurement, TOF050C, Register Map

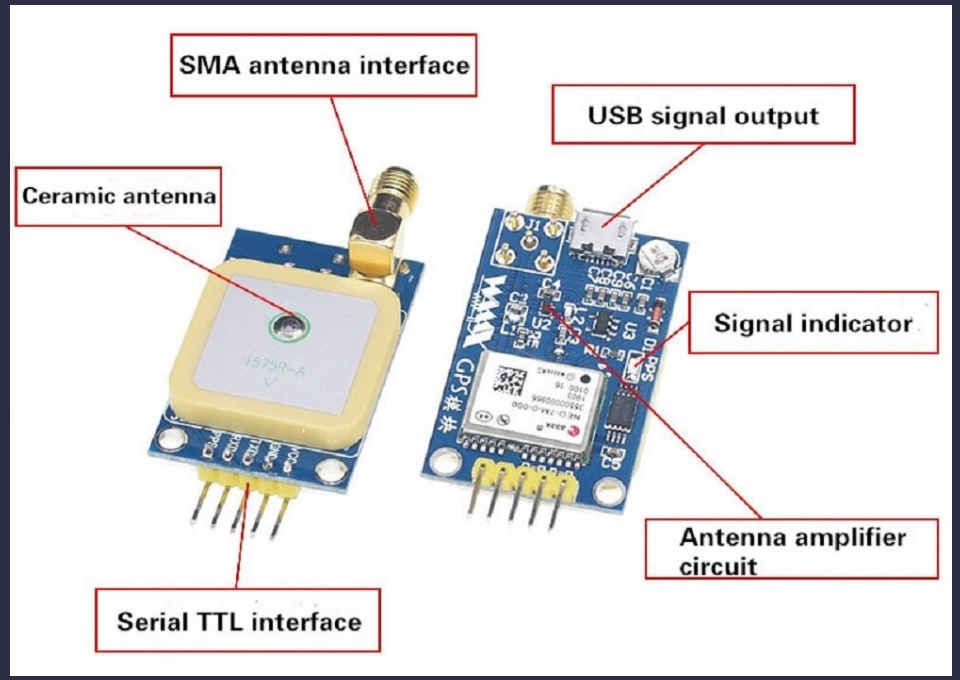
Offset	Register name
0x03E	SYSALS__INTERMEASUREMENT_PERIOD
0x03F	SYSALS__ANALOGUE_GAIN
0x040	SYSALS__INTEGRATION_PERIOD
0x04D	RESULT__RANGE_STATUS
0x04E	RESULT__ALS_STATUS
0x04F	RESULT__INTERRUPT_STATUS_GPIO
0x050	RESULT__ALS_VAL
0x052:0x060 (0x2)	RESULT__HISTORY_BUFFER_x
0x062	RESULT__RANGE_VAL
0x064	RESULT__RANGE_RAW
0x066	RESULT__RANGE_RETURN_RATE
0x068	RESULT__RANGE_REFERENCE_RATE
0x06C	RESULT__RANGE_RETURN_SIGNAL_COUNT
0x070	RESULT__RANGE_REFERENCE_SIGNAL_COUNT
0x074	RESULT__RANGE_RETURN_AMB_COUNT
0x078	RESULT__RANGE_REFERENCE_AMB_COUNT
0x07C	RESULT__RANGE_RETURN_CONV_TIME
0x080	RESULT__RANGE_REFERENCE_CONV_TIME
0x10A	READOUT__AVERAGING_SAMPLE_PERIOD
0x119	FIRMWARE__BOOTUP
0x120	FIRMWARE__RESULT_SCALER
0x212	I2C_SLAVE__DEVICE_ADDRESS
0x2A3	INTERLEAVED_MODE__ENABLE

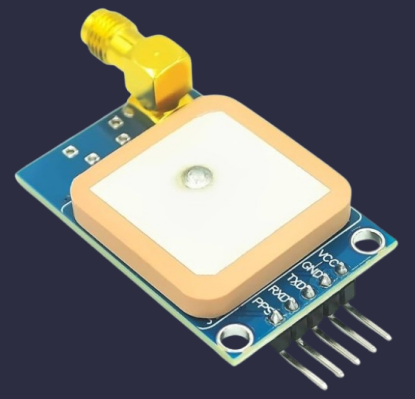
System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - Laser Distance Measurement, TOF050C
  - Measurement
  - Write SYSRANGE_START (Addr 0x0) to 1
  - Wait RESULT_INTERRUPT_STATUS Register 0x4F
  - Read RESULT_RANGE_VAL Register 0x62

System Design with Sensors I - Baremetal

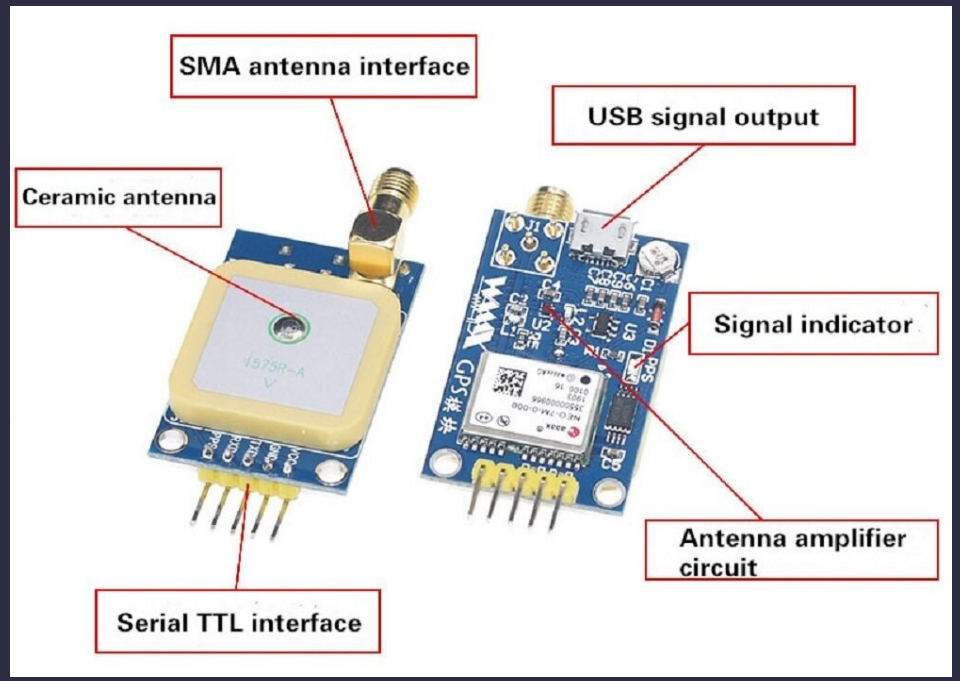
- Sensor Development Boards
  - Sensors
  - GPS, NEO-7M
  - GPS Satellite Positioning Module
  - UART Interface
  - NMEA (National Marine Electronics Association) 0183
  - Details:
  - https://content.u-blox.com/sites/default/files/products/documents/NEO-7_DataSheet_%28UBX-13003830%29.pdf

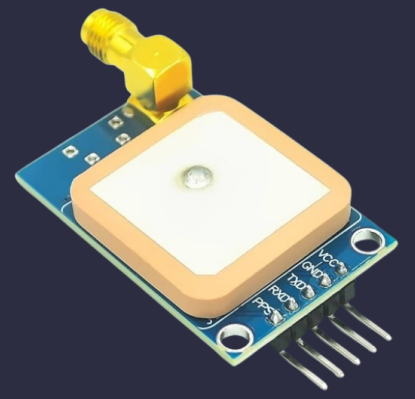




System Design with Sensors I - Baremetal

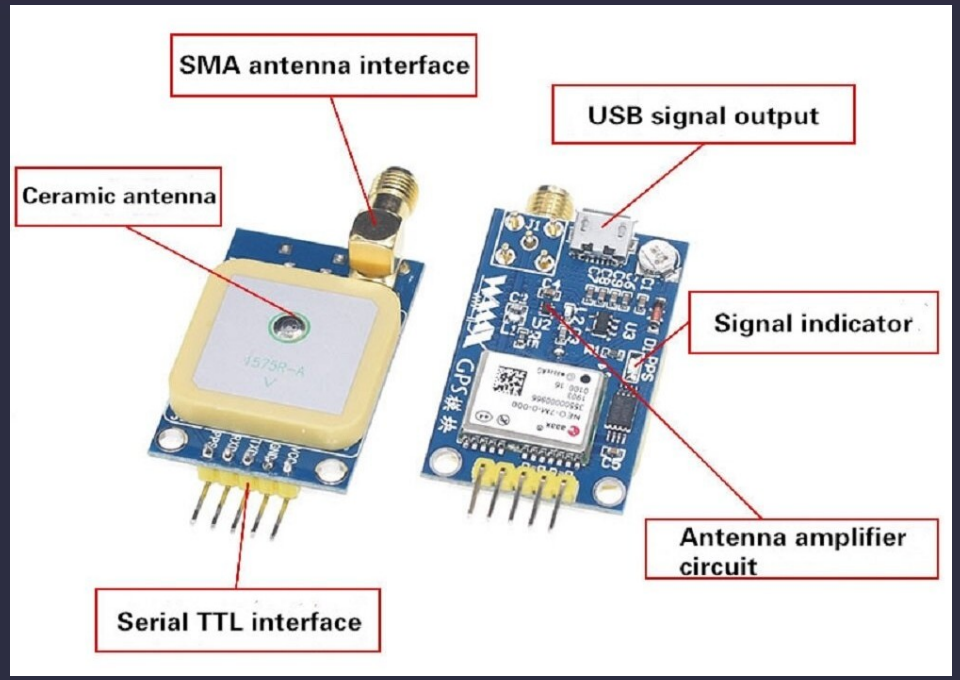
- Sensor Development Boards
  - Sensors
  - GPS, NEO-7M
  - GPRMC (Recommended Minimum Navigation Information):Provides essential data including time, date, latitude, longitude, speed, and course over ground.
  - GPGGA (Global Positioning System Fix Data):Contains detailed fix information such as latitude, longitude, altitude, fix quality, and the number of satellites used.
  - GPGSA (GPS DOP and Active Satellites):Indicates which satellites are being used for the fix and provides dilution of precision (DOP) values for positioning accuracy.
  - GPGSV (GPS Satellites in View):Lists all satellites in view along with details like satellite ID, elevation, azimuth, and signal-to-noise ratio.
  - GPVTG (Track Made Good and Ground Speed):Reports the ground track (direction) and speed over ground.
  - GPGLL (Geographic Position – Latitude/Longitude):Outputs the current position (latitude and longitude) along with a status indicator to show if the data is valid.

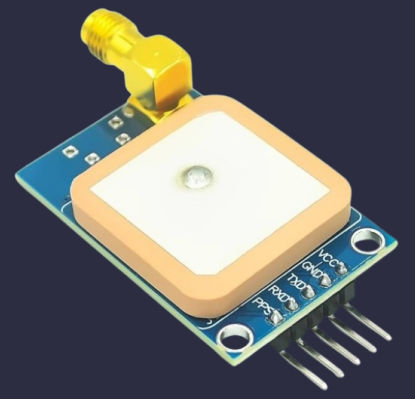




System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - GPS, NEO-7M
  - Sample NMEA Message

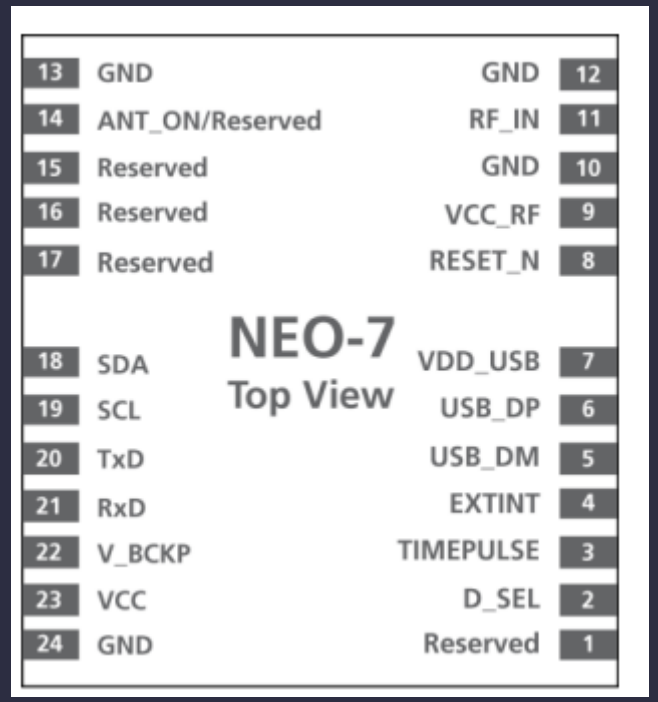


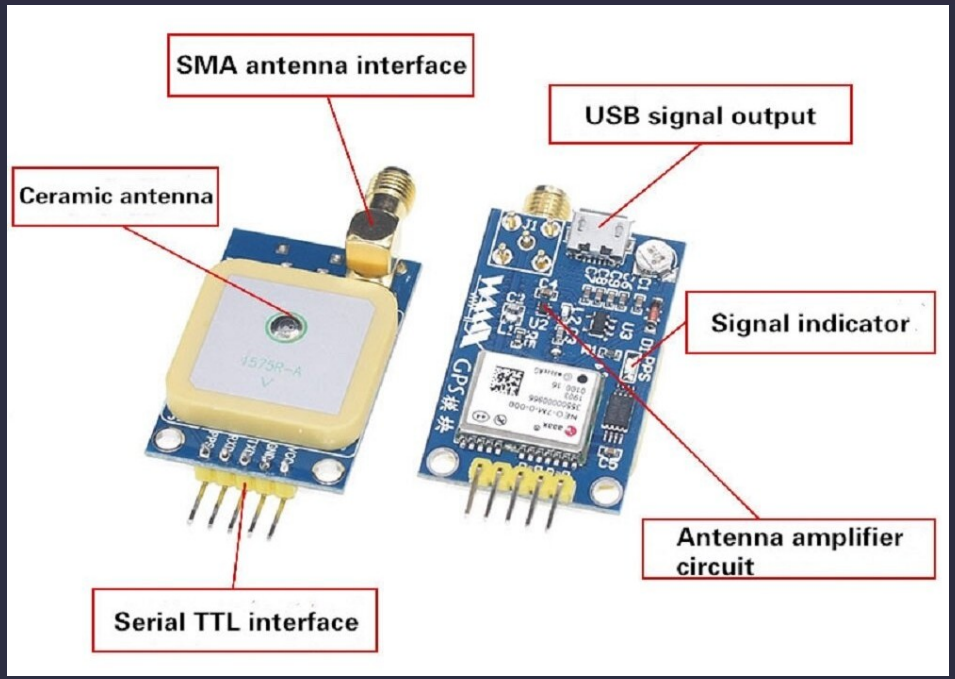


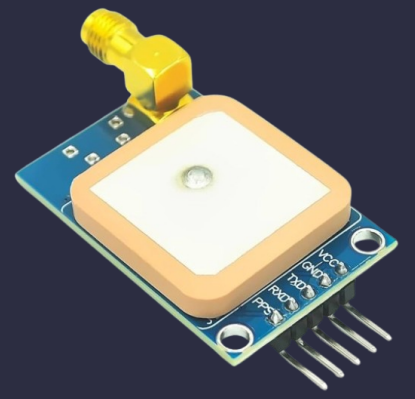
```
Text Console
$GBGSV,3,3,11,27,58,131,33,20,16,217,28,32,72,199,34,8*4D
$GBGSV,1,1,04,05,38,141,34,13,22,059,27,02,22,114,35,09,14,107,28,B*07
$GAGSV,2,1,05,33,60,068,32,10,26,299,24,12,49,312,30,19,44,156,27,1*79
$GAGSV,2,2,05,26,16,099,29,1*48
$GAGSV,2,1,05,33,60,068,33,10,26,299,26,12,49,312,32,19,44,156,28,2*74
$GAGSV,2,2,05,26,16,099,25,2*47
$GAGSV,1,1,04,33,60,068,34,12,49,312,36,19,44,156,30,31,12,223,25,7*7F
$GNGGA,115144.00,3953.37121850,N,03243.10270985,E,2,06,2.2,955.0549,M,37.3246,M,02,0001*77
$GNZDA,115144.00,12,06,2024,,*7D
$GNGSA,M,3,08,21,27,,,,,,,,,6.9,2.2,6.6,1*3C
$GNGSA,M,3,65,72,88,,,,,,,,,6.9,2.2,6.6,2*37
$GNRMC,115144.00,A,3953.37121850,N,03243.10270985,E,1.197,242.8,120624,4.5,E,D,V*51
$GNGST,115144.00,1.07,24.68,8.92,59.1708,11.701,17.158,30.796*75
$GNVTG,242.808,T,238.325,M,1.19680,N,2.21647,K,A*37
$GPGSV,2,1,05,08,61,274,38,10,56,040,33,21,37,305,35,27,59,192,34,1*62
$GPGSV,2,2,05,23,22,053,29,1*5D
$GPGSV,1,1,02,10,56,040,26,21,37,305,33,4*60
$GPGSV,1,1,04,08,61,274,34,10,56,040,30,27,59,192,33,23,22,053,25,8*66
$GLGSV,1,1,03,65,63,195,35,88,55,338,35,72,51,038,34,1*45
$GLGSV,1,1,03,65,63,195,28,88,55,338,33,72,51,038,29,3*41
$GBGSV,2,1,07,30,65,309,34,41,46,048,29,60,24,119,24,13,22,059,27,1*78
$GBGSV,2,2,07,27,58,131,32,20,16,217,25,32,72,199,33,1*48
```

System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - GPS, NEO-7M, I/Os

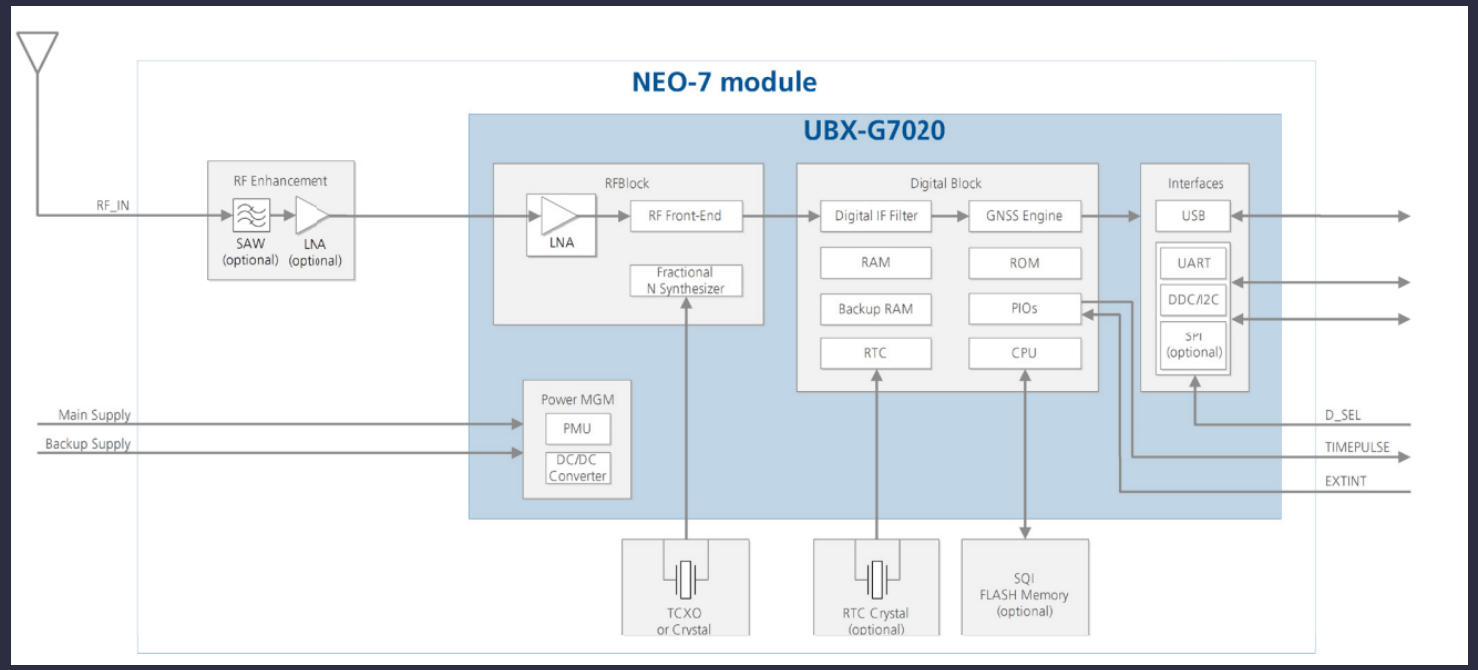


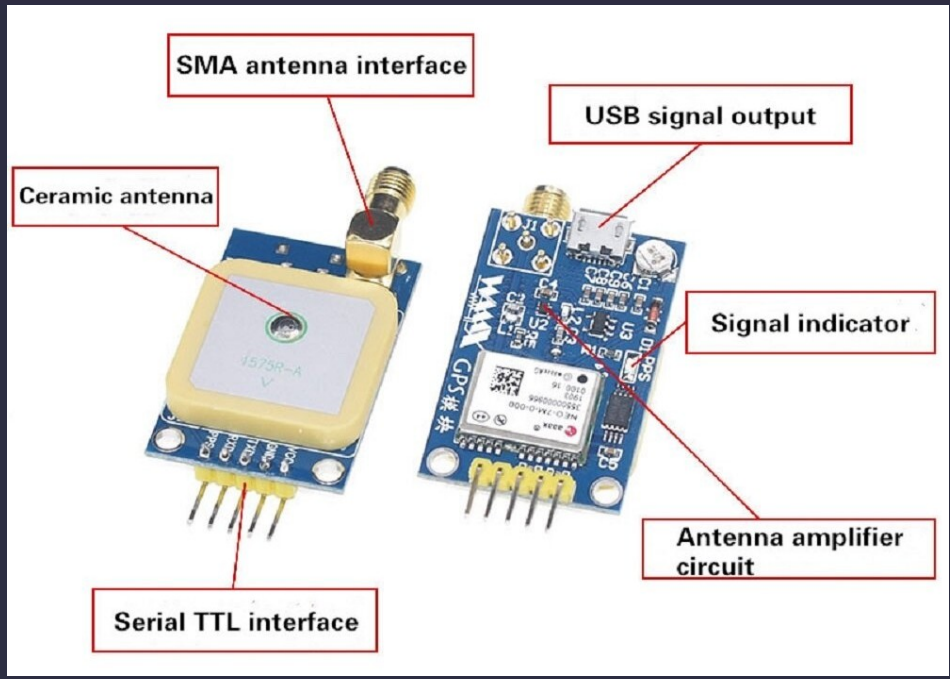


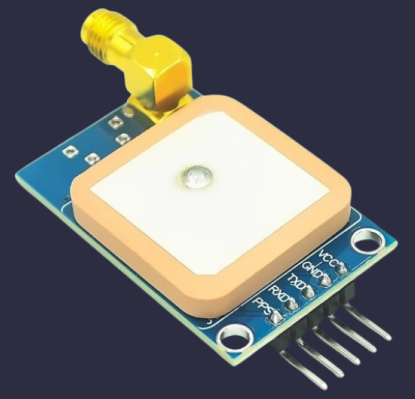


System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - GPS, NEO-7M, Block Diagram

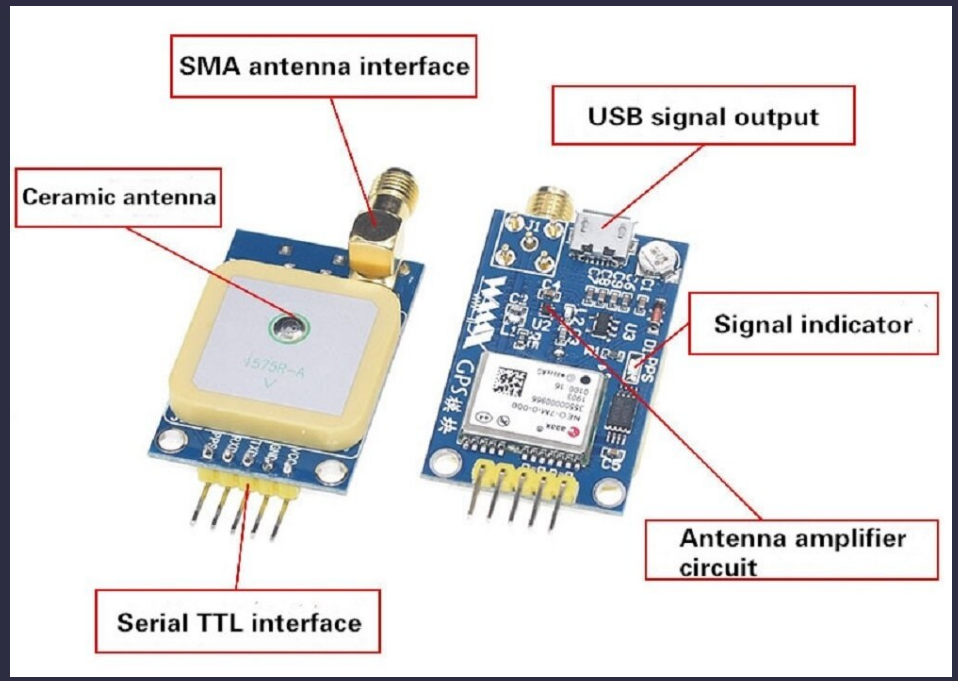


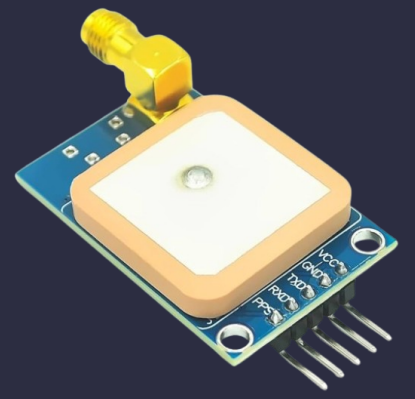




System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - GPS, NEO-7M, Example Settings
  - Baud Rate Change: Can change the default baud rate (e.g., from 9600 to 115200) using the UBX-CFG-PRT command.
  - Update Rate Adjustment: Configure the measurement rate (e.g., setting the module to output data at 1 Hz or 5 Hz) with the UBX-CFG-RATE command.
  - NMEA Sentence Control: Enable or disable specific NMEA sentences (like turning off the GPGSV sentence) using UBX-CFG-MSG or similar commands to reduce data load.
  - Dynamic Model Setting: Adjust the dynamic model (e.g., for pedestrian, automotive, or airborne use) via the UBX-CFG-NAV5 command.





System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors

  - RF Transceiver, Dorji DRF1278DM

  - Semtech SX1278 IC
  - Voltage: 3,4V-5,5V
  - Communication: UART
  - Maximum Power: 20dBm - 100mW
  - Receiver Sensivity 300bps: -138dBm
  - Data Rate: 1,2Kbps - 9,6Kbps - 57,6Kbps
  - Frequency 420MHz - 450MHz
  - Rf Wakeup Time: 2sec - 10sec

  - Details:
  https://www.semtech.com/products/wireless-rf/lora-connect/sx1278

SEMTECH
SX1278

System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - RF Transceiver, Dorji DRF1278DM, Block Diagram

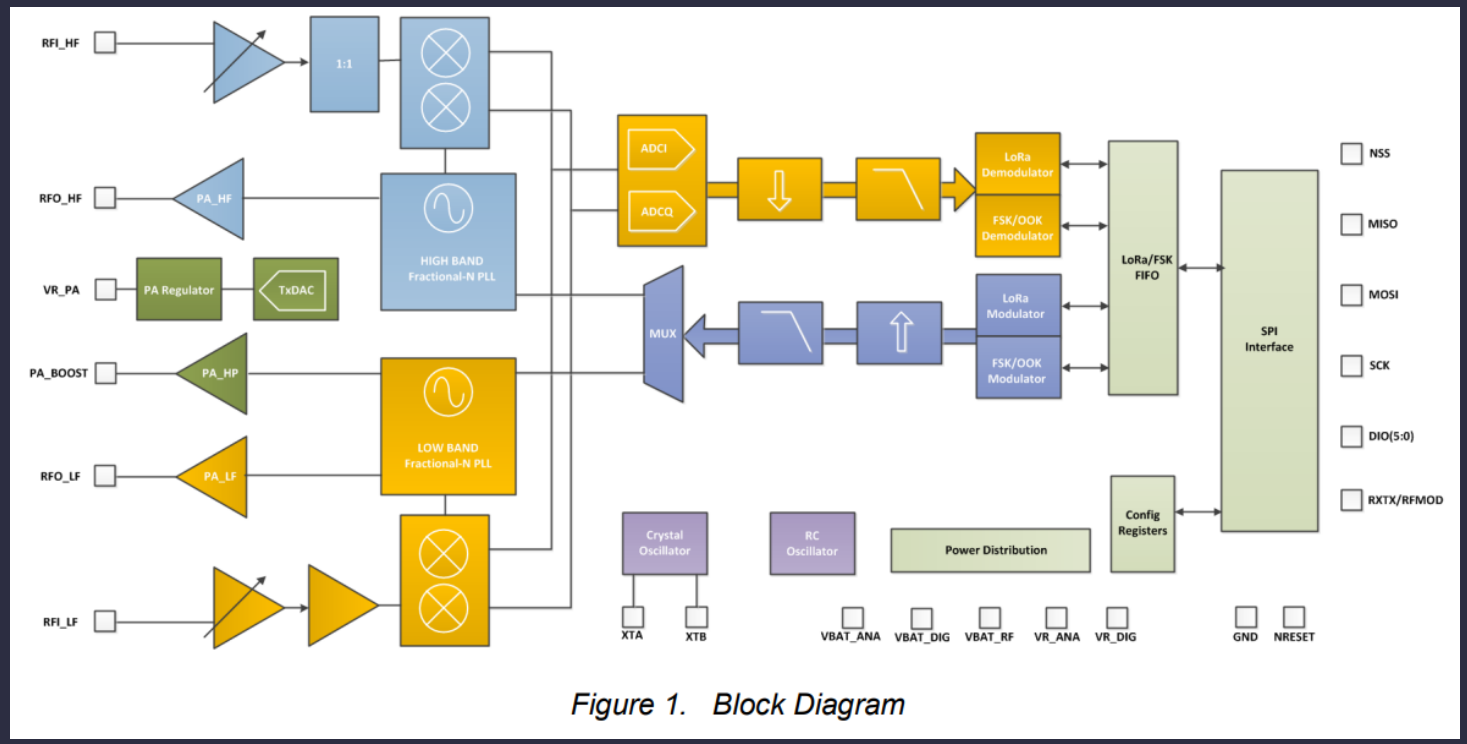
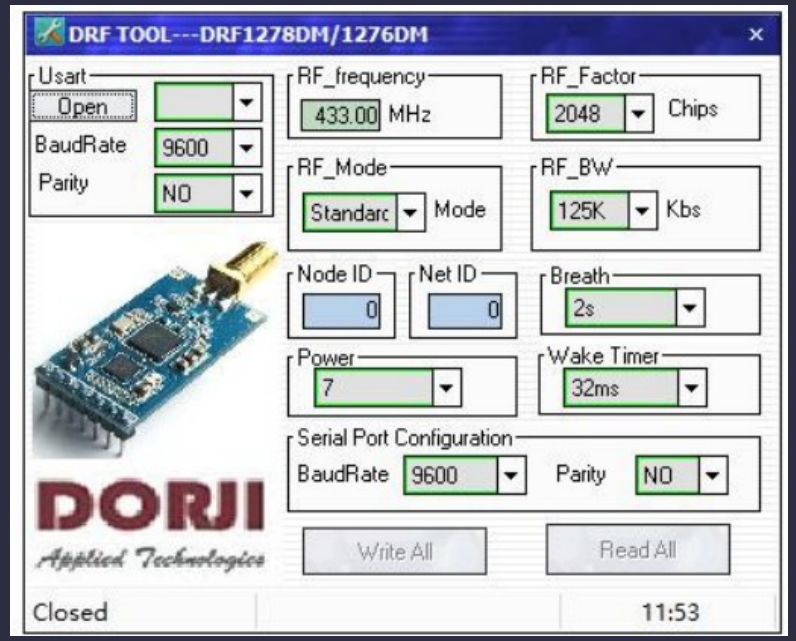


Figure 1. Block Diagram

System Design with Sensors I - Baremetal

- Sensor Development Boards
  - Sensors
  - RF Transceiver, Dorji DRF1278DM, Block Diagram



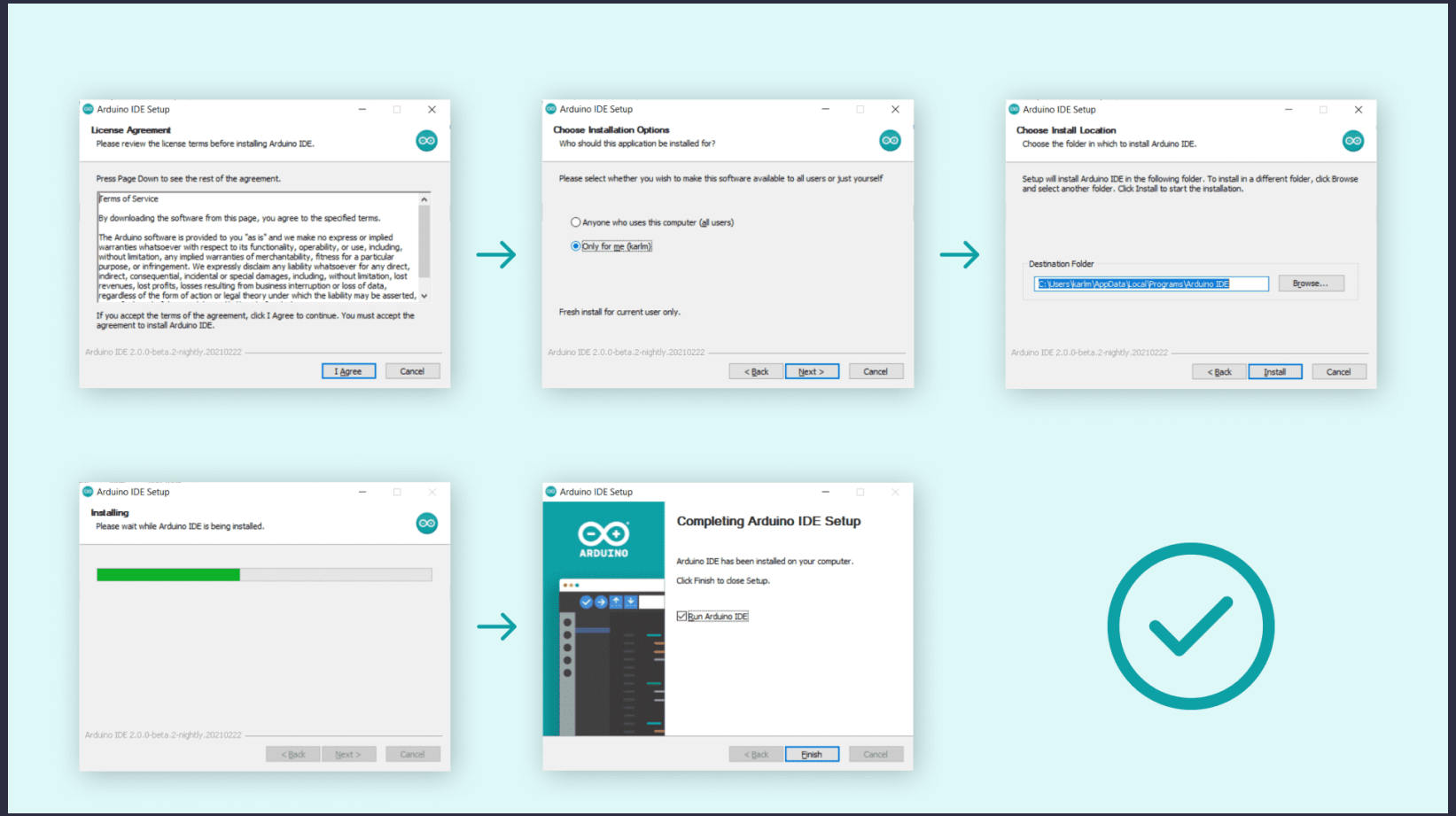
- Details:
https://www.dorji.com/docs/data/DRF1278DM.pdf

System Design with Sensors I - Baremetal

- Ardunio IDE

Download and Install

https://www.arduino.cc/en/software#experimental-software

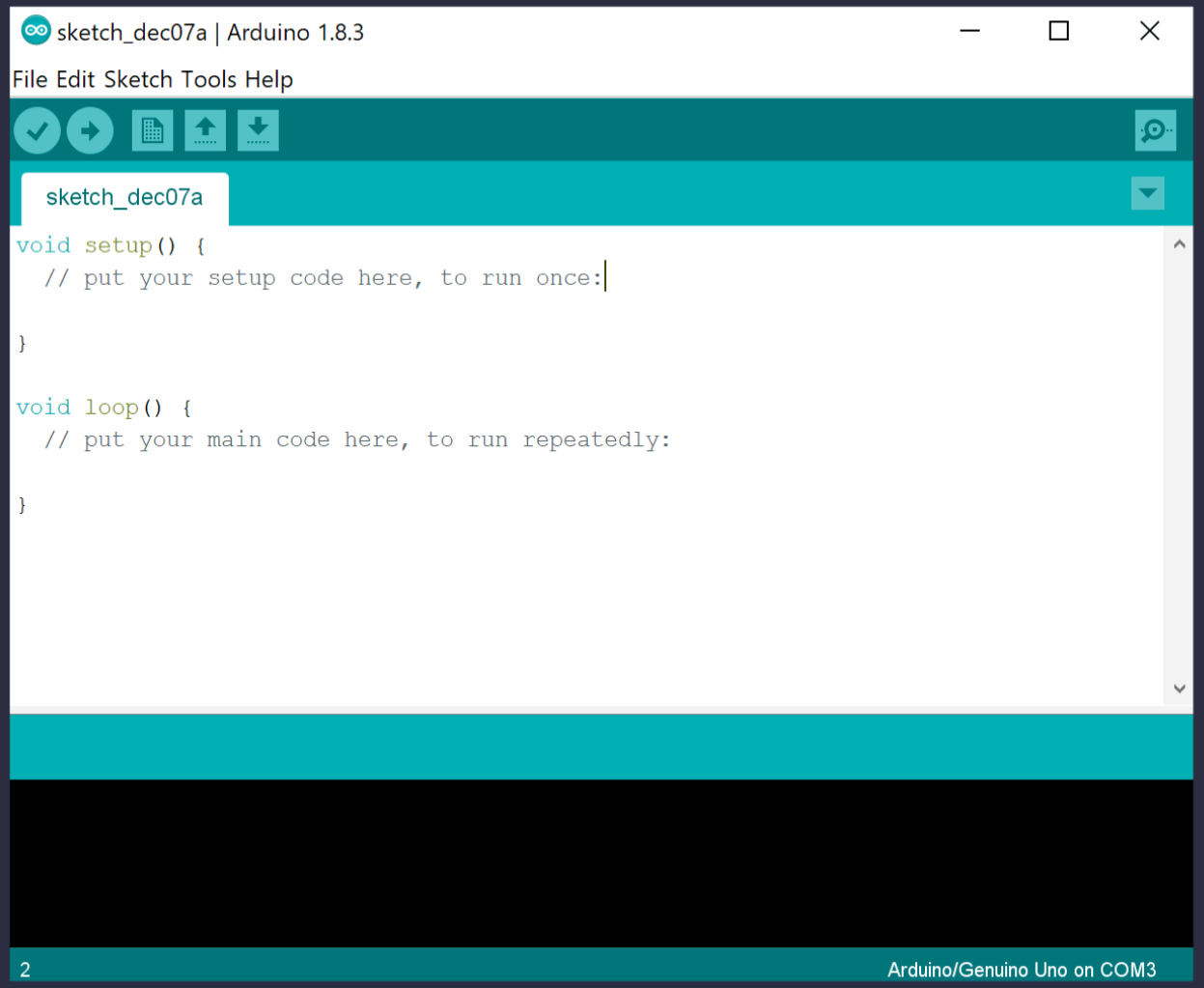


System Design with Sensors I - Baremetal

- Ardunio IDE

There are two main tools when uploading a sketch to a board: verify and upload.

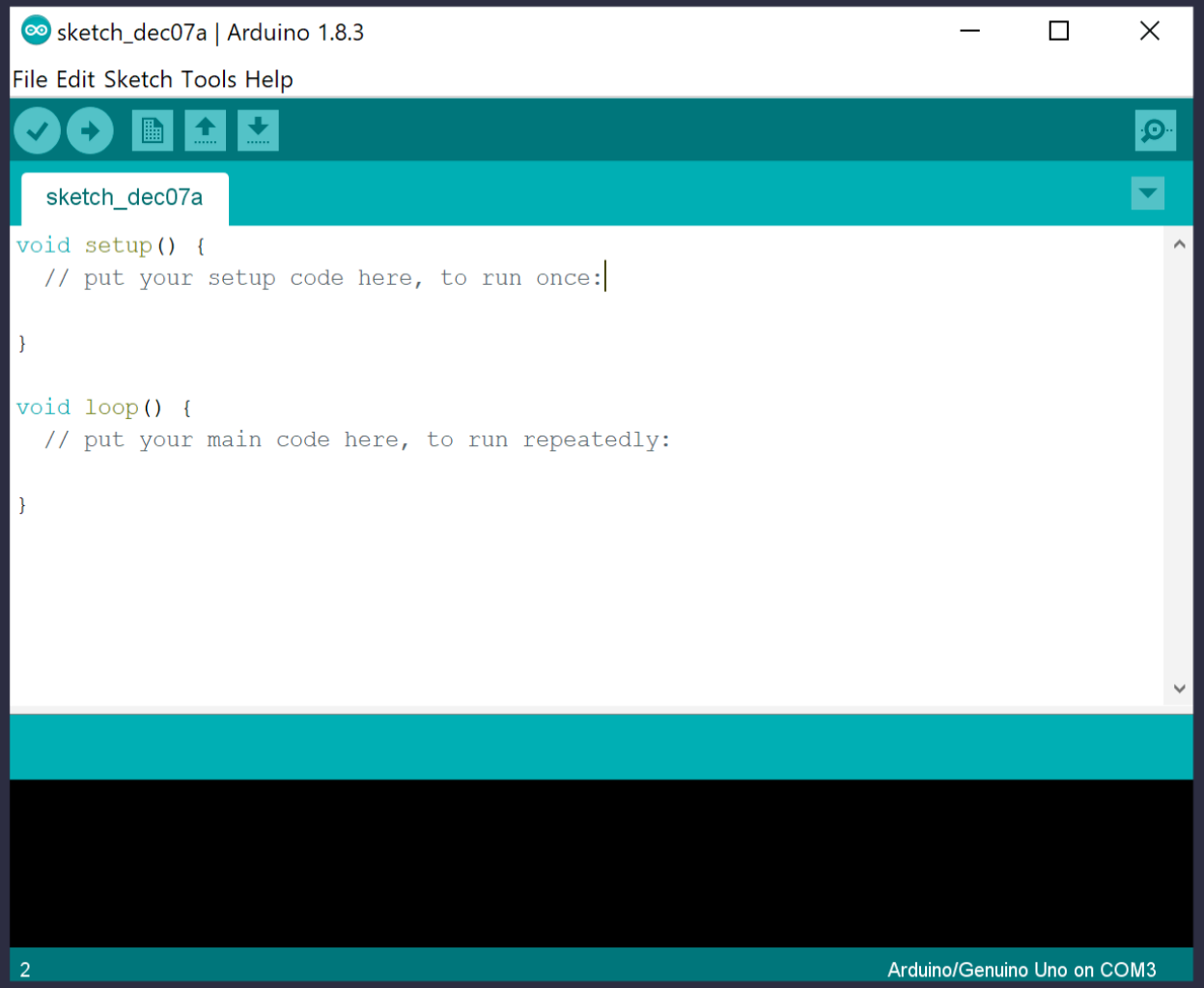
- The verify tool simply goes through your sketch, checks for errors and compiles it.
- The upload tool does the same, but when it finishes compiling the code, it also uploads it to the board



System Design with Sensors I - Baremetal

- Ardunio IDE

A good practice is to use the verifying tool before attempting to upload anything. This is a quick way of spotting any errors in your code, so you can fix them before actually uploading the code.



System Design with Sensors I - Baremetal

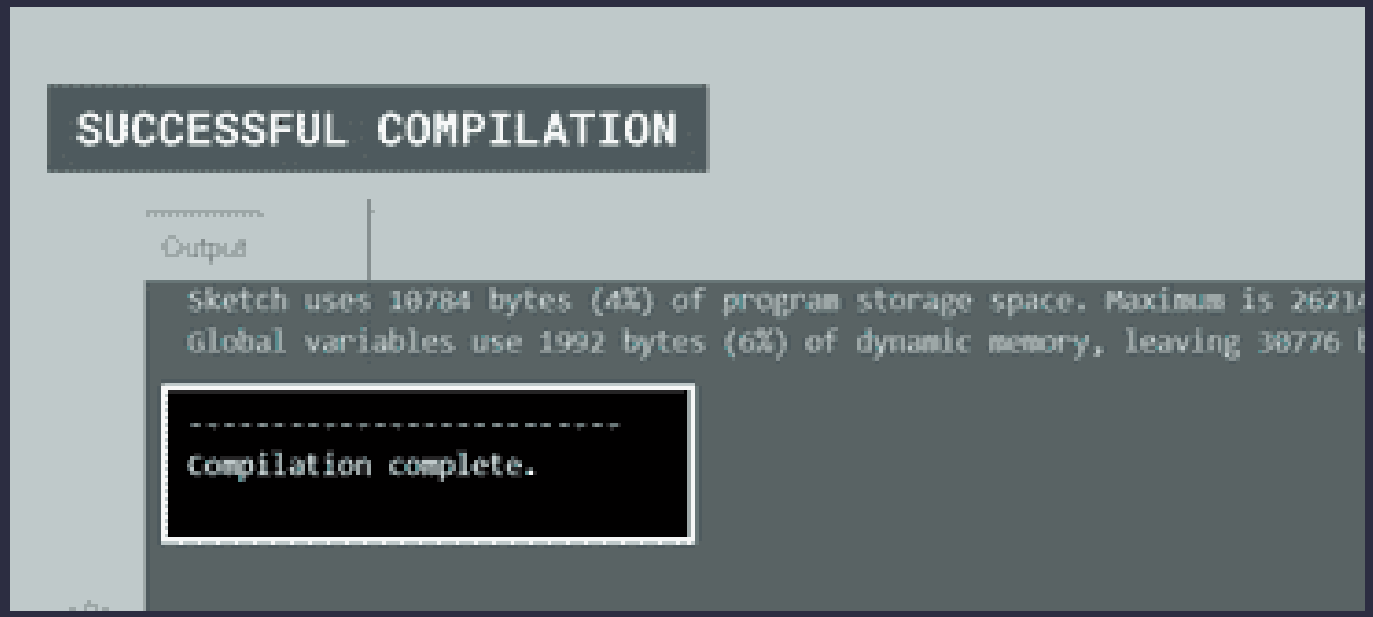
- Ardunio IDE

At the very left, there is a checkmark and an arrow pointing right. The checkmark is used to verify, and the arrow is used to upload.

System Design with Sensors I - Baremetal

- Ardunio IDE

Click on the verify tool (checkmark).

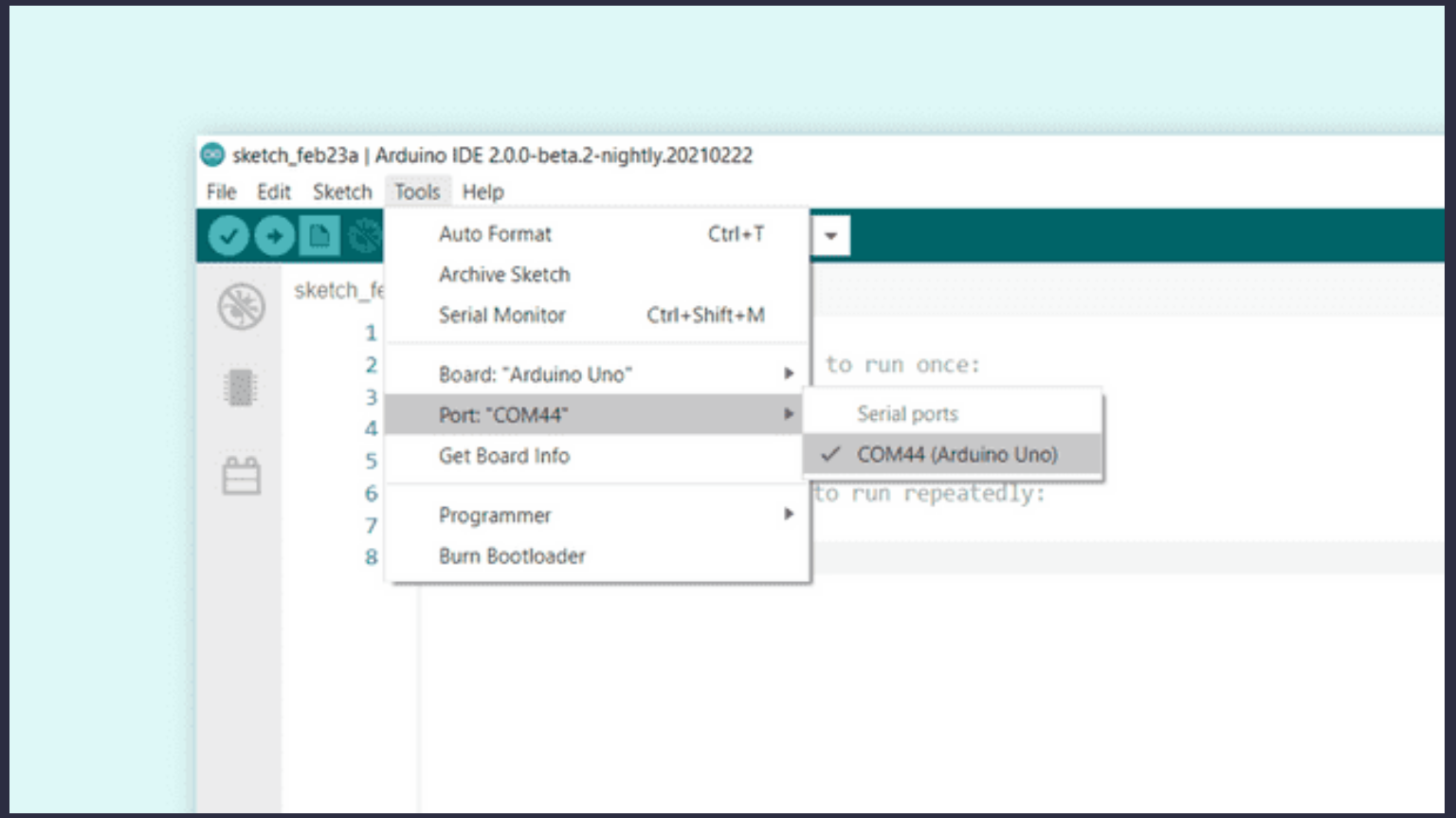


System Design with Sensors I - Baremetal

- Ardunio IDE

Select the board that we are using

Tools > Port > {Board}

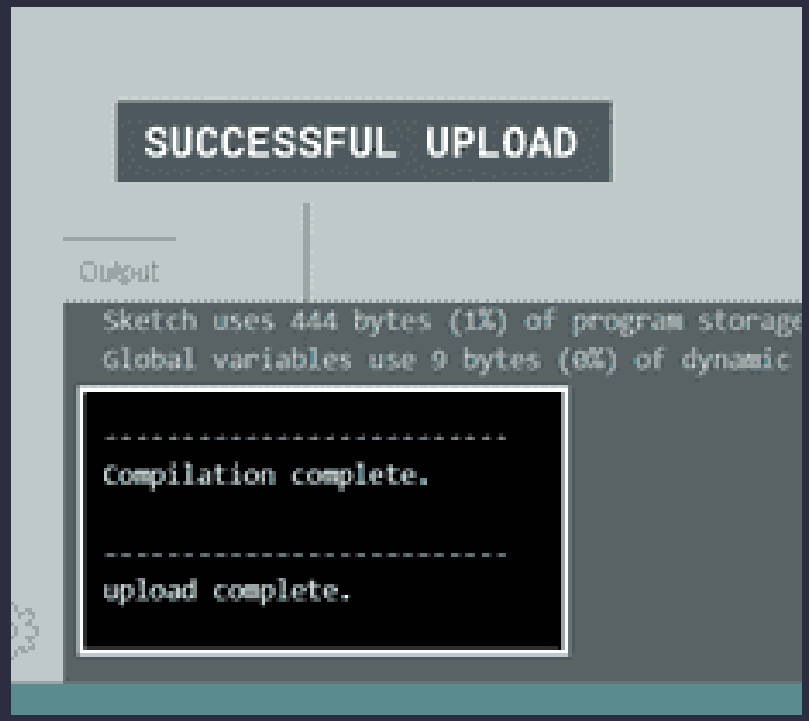


System Design with Sensors I - Baremetal

- Ardunio IDE

Click on the upload button, and it will start uploading the sketch to the board

When it is finished, it will notify you in the console log.

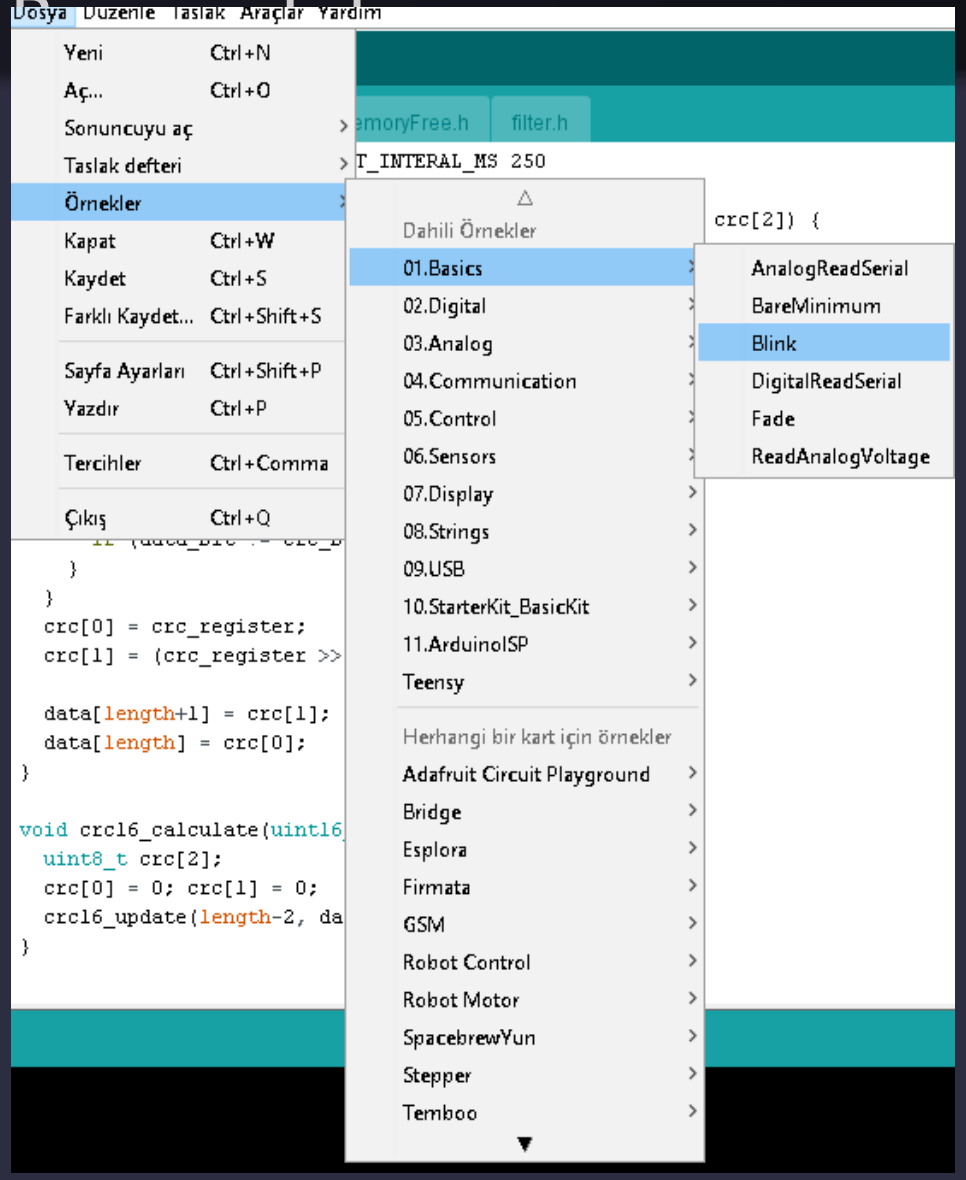


System Design with Sensors I - D...

- Ardunio IDE

Examples under

File -> Examples ->

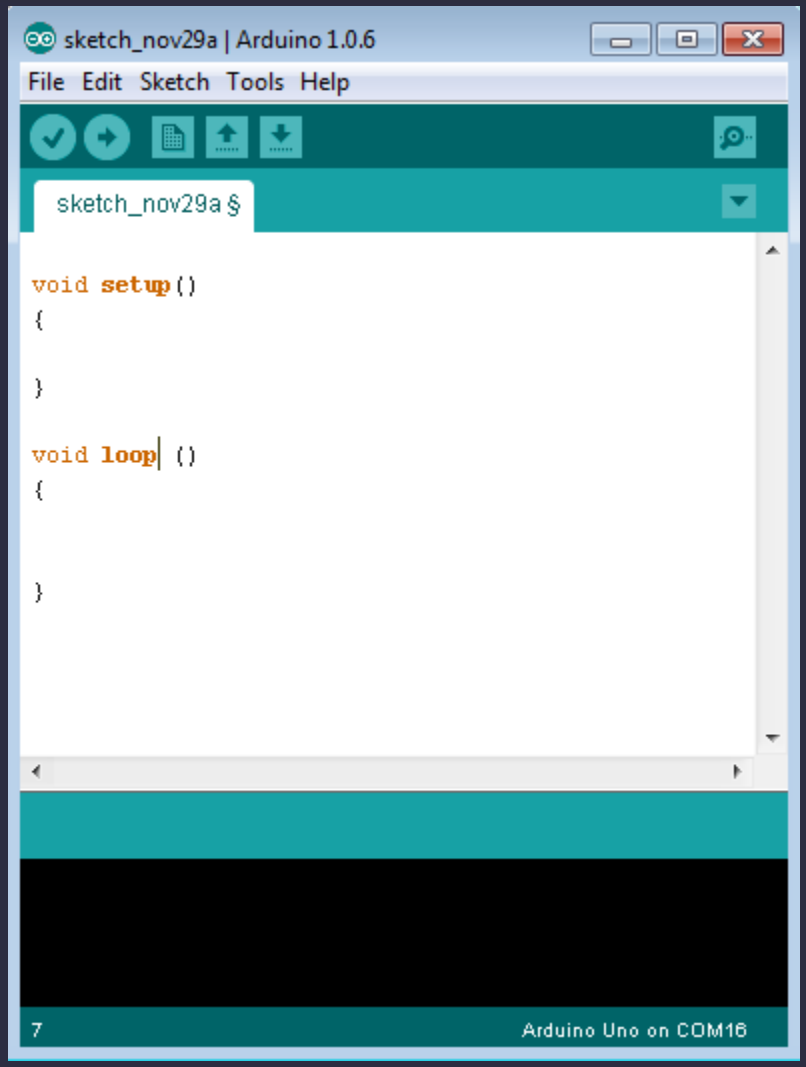


System Design with Sensors I - Baremetal

- Ardunio IDE

Software structure consist of two main functions

- Setup( ) function
- Loop( ) function



System Design with Sensors I - Baremetal

- Ardunio IDE

```
void setup ( ) {

}
```

PURPOSE – The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

System Design with Sensors I - Baremetal

- Ardunio IDE

```
void loop ( ) {

}
```

PURPOSE – After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

System Design with Sensors I - Baremetal

- Ardunio IDE

Data types

void	**Boolean**	**char**	**Unsigned char**	**byte**	**int**	**Unsigned int**	**word**
long	Unsigned long	short	float	double	array	String-char array	String-object

System Design with Sensors I - Baremetal

- Ardunio IDE

The pins on the Arduino board can be configured as either inputs or outputs.

pinMode() function can set a pin to input or output

pinMode(3,INPUT) ; // set pin to input without using built in pull up resistor

System Design with Sensors I - Baremetal

- Ardunio IDE

Arduino provides four different time manipulation functions

	Function & Description
1	delay () function The way the **delay()** function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds).
2	delayMicroseconds () function The **delayMicroseconds()** function accepts a single integer (or number) argument. There are a thousand microseconds in a millisecond, and a million microseconds in a second.
3	millis () function This function is used to return the number of milliseconds at the time, the Arduino board begins running the current program.
4	micros () function The micros() function returns the number of microseconds from the time, the Arduino board begins running the current program. This number overflows i.e. goes back to zero after approximately 70 minutes.

System Design with Sensors I - Baremetal

- Ardunio IDE

digitalWrite() Function

The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode()

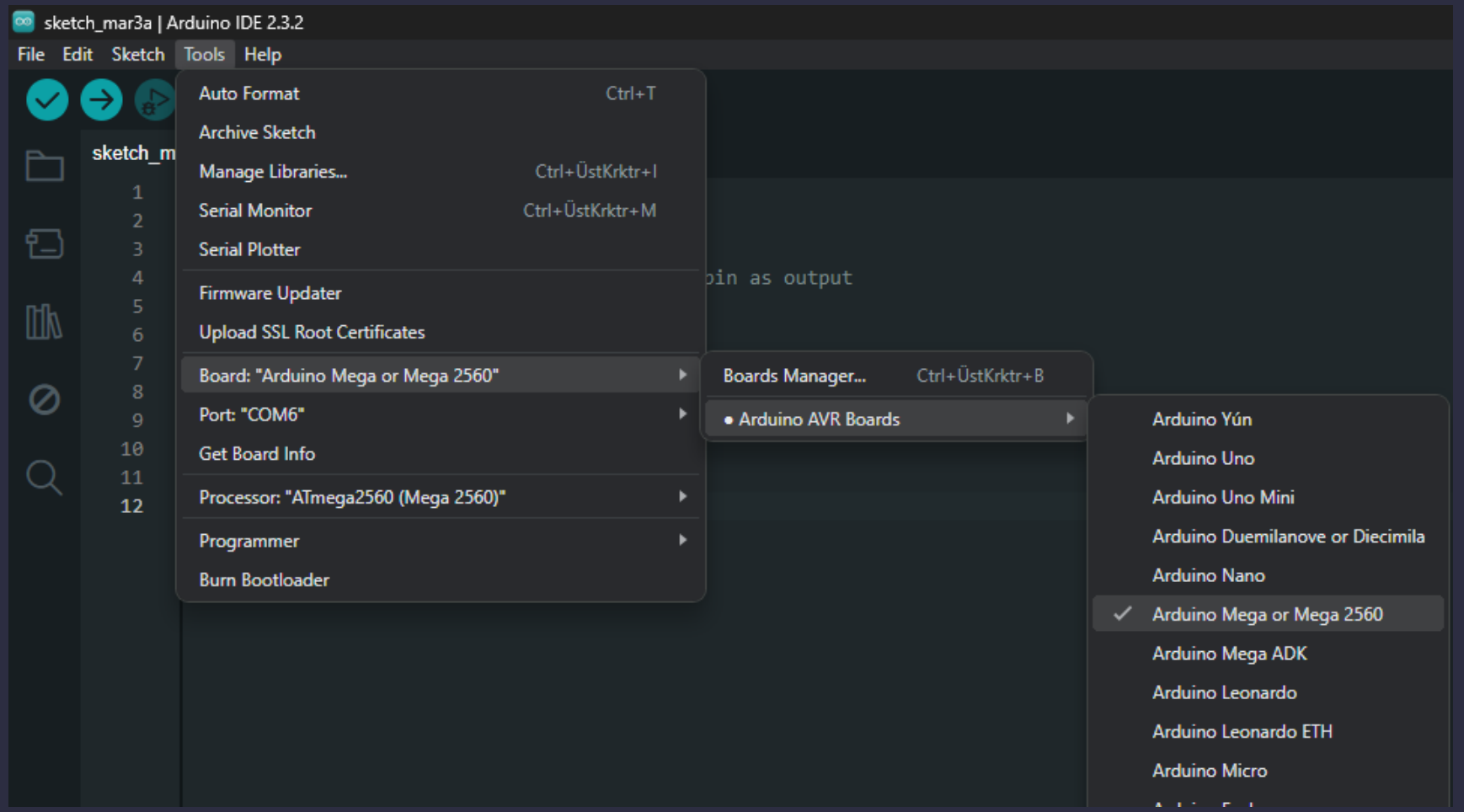
its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

System Design with Sensors I - Baremetal

- Ardunio IDE

Compilation

Select Board



System Design with Sensors I - Baremetal

- Ardunio IDE

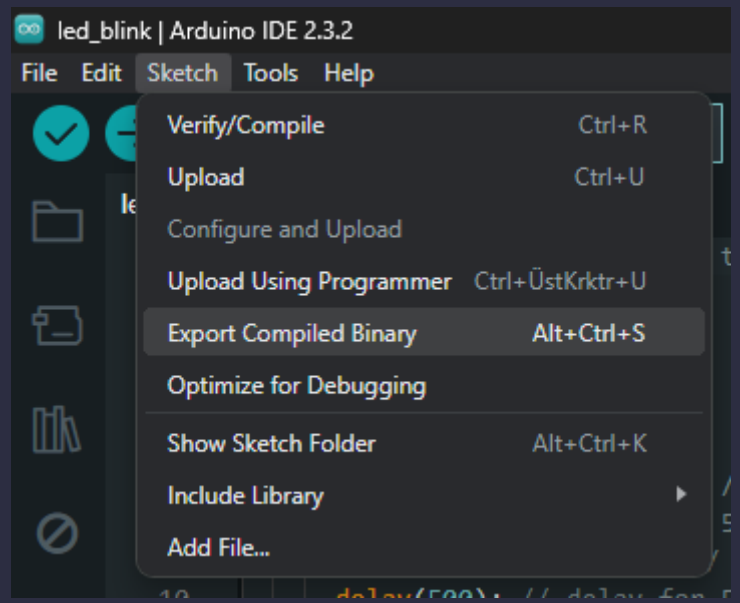
```
int led = 5 ; // led connected to pin 5

void setup () {
  pinMode(led , OUTPUT);
}
void loop () {
     digitalWrite(led,HIGH); // turn on led
    delay(500); // delay for 500 ms
    digitalWrite(led,LOW); // turn off led
    delay(500); // delay for 500 ms
}
```

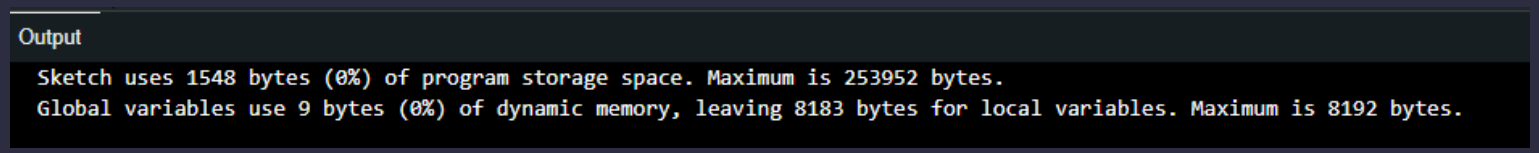
Save the Code

System Design with Sensors I - Baremetal

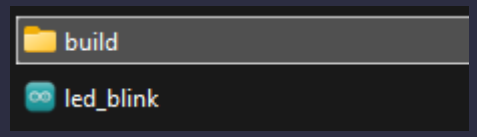
- Ardunio IDE

Compile and Export as Hex





No Errors



build folder will be created