Interfacing Sensors & Modules to Microcontrollers

- Presentation Topics

I. Microprocessors & Microcontroller
II. Hardware/software Tools for Interfacing
III. Type of Sensors/Modules
IV. Level Inputs (Digital ON/OFF)
V. Example 1: Interfacing Random Pulses From Radiation Detector
VI. Example 2: Interfacing Pulse Inputs with Coded Information
VII. Synchronous & Asynchronous Communication
VIII. Using Bluetooth SPP with Virtual Com Ports, Android Cell Phones/Tablets
IX. Interfacing Motion Sensing Devices
X. Example 3: Wing Control Actuating System – Catastrophe Avoidance

Microprocessors/Microcontrollers

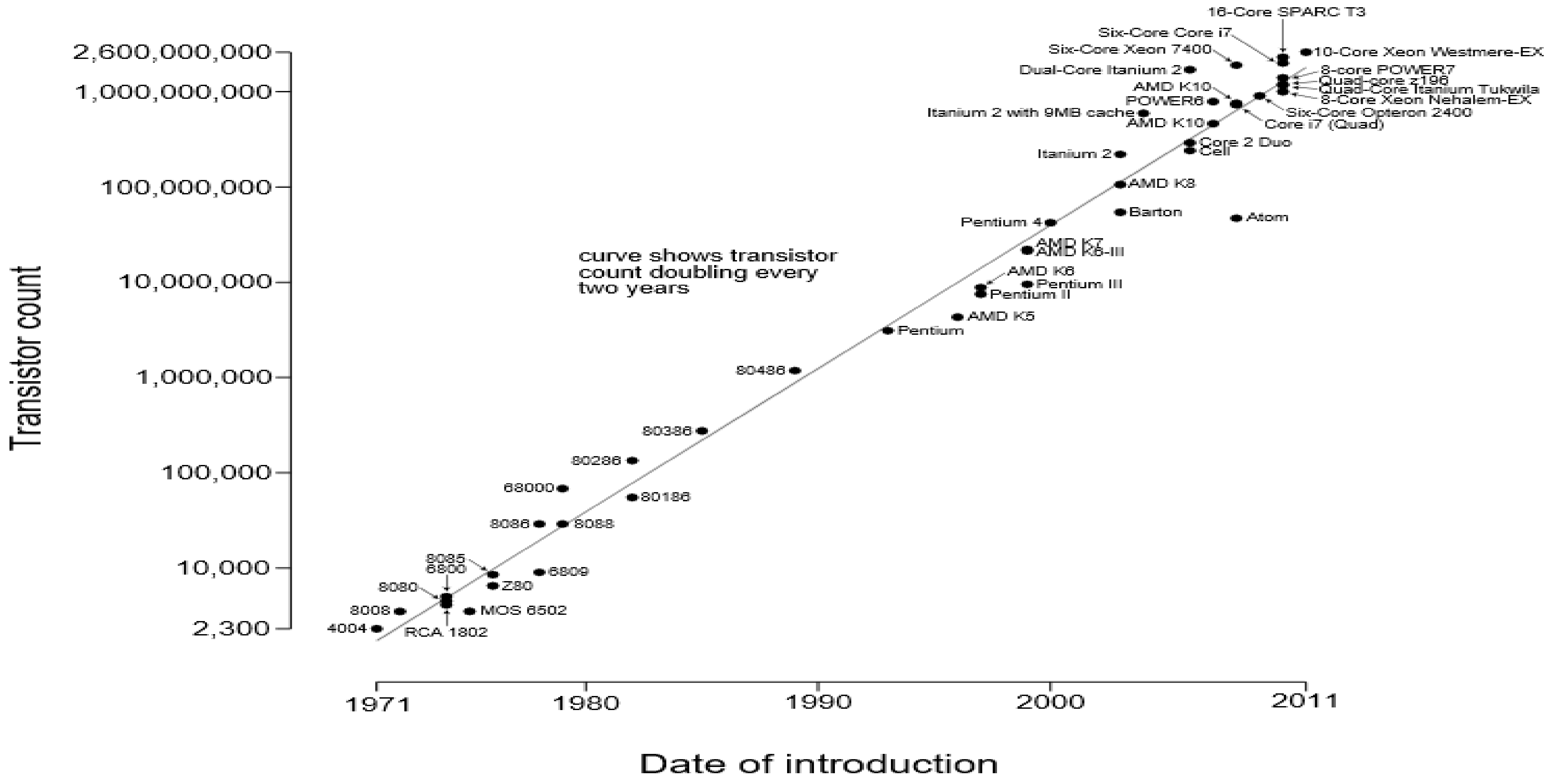
- The first complete single-chip **microprocessor**, Intel's 4004, was introduced in 1971

- **Gary Boone** of Texas Instruments was working on quite a similar concept and invented the **microcontroller**- **TMS1802NC**

- Microprocessor- a central processor on a chip
  - Building block to create a computing devices
  - ROM, RAM, I/O Ports, decoding logic are added to the bus system

- Microcontroller - a chip that contains a central processor plus RAM, ROM, I/O Ports
  - Microcontrollers are a complete computing/processing system
  - Can be programmed in assembler, C, and in many high-level languages
  - Interfacing involves attaching I/O devices (sensors and modules) to I/O Ports
  - The internal bus system is not available to attach I/O devices

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Interfacing Sensors & Modules to Microcontrollers

GENERALLY REQUIRES
SOFTWARE/HARDWARE
TOOLS

USEFUL TOOLS

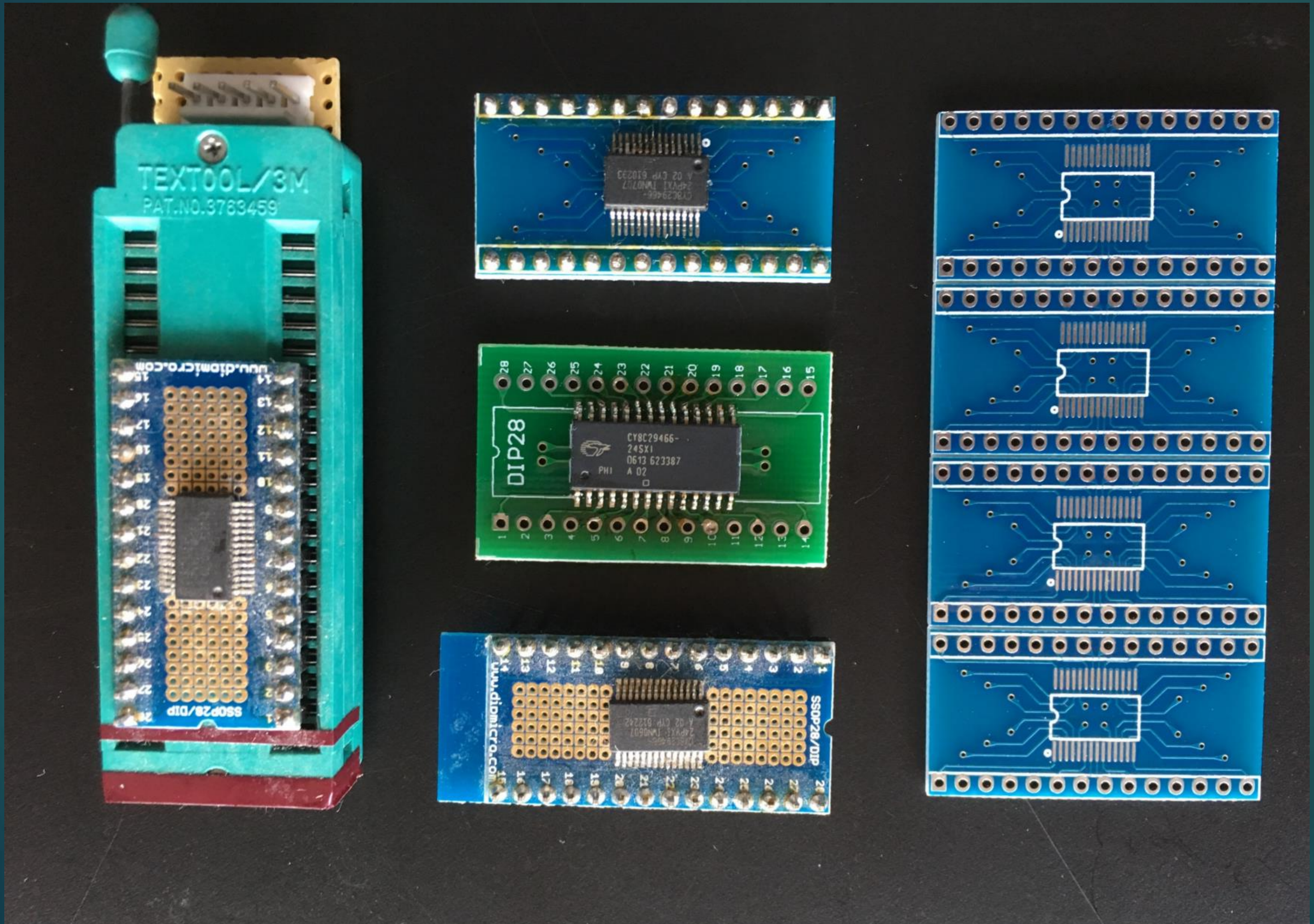
Hardware Tools

- Breadboards – come in all shape and Sizes
- Adapter PCB converter boards – available for most MCU's footprints
- PCB boards designed for specific MCU – Eagle Software
- Temperature controlled soldering iron – SMD devices
- Wire wrap Gun
- Hot Air rework gun
- Digital multimeter
- Oscilloscope

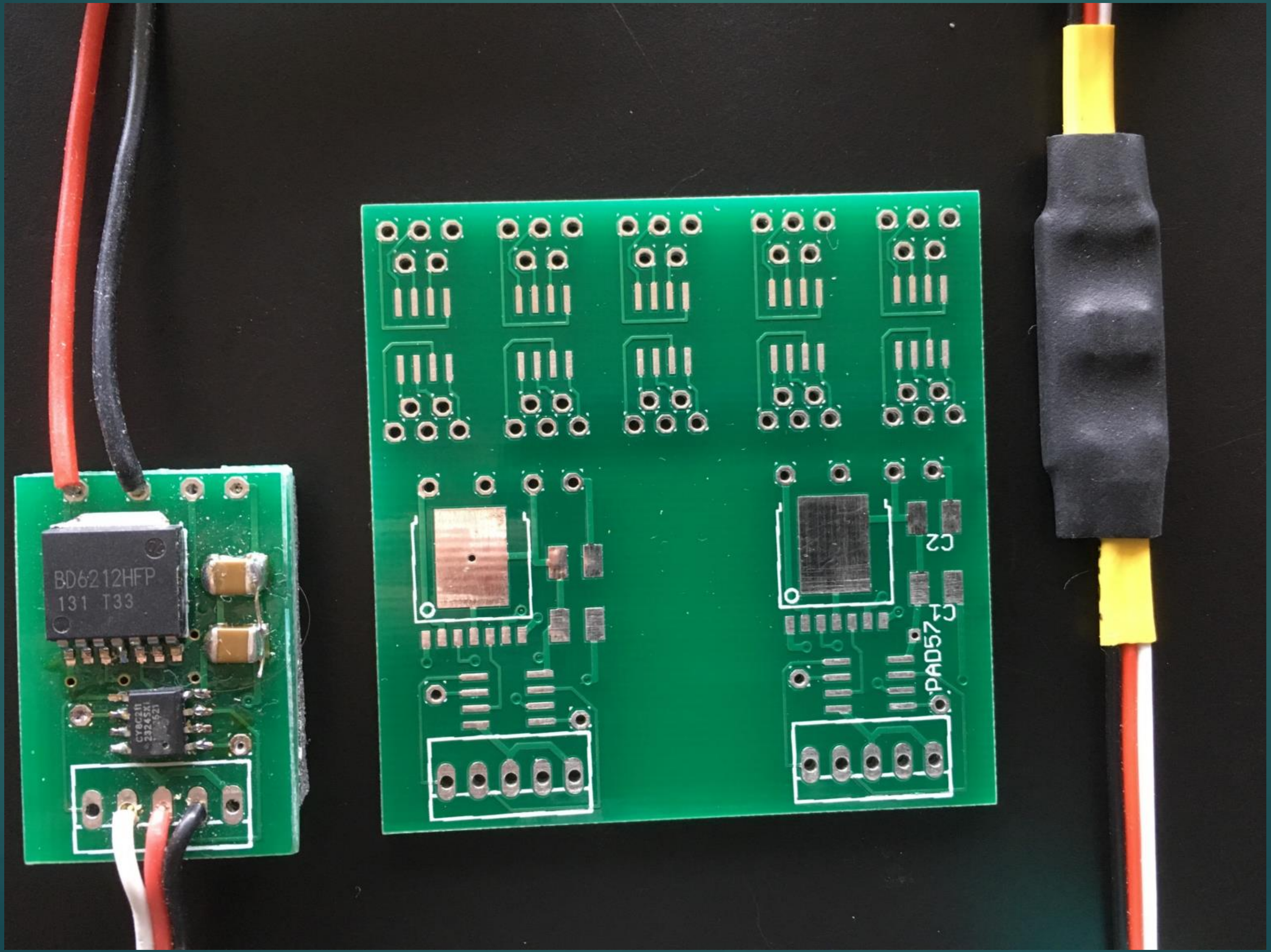
Software Tools

- Eagle Software – creating PCB boards
  - Limited to 2 schematic sheets, 2 signal layers, and 80 cm^2 board area
- Tera Term – Terminal emulator ASCII serial communication
- Realterm - Serial and TCP terminal for engineering and debugging
- Bluetooth SPP Pro – android phone/tablet

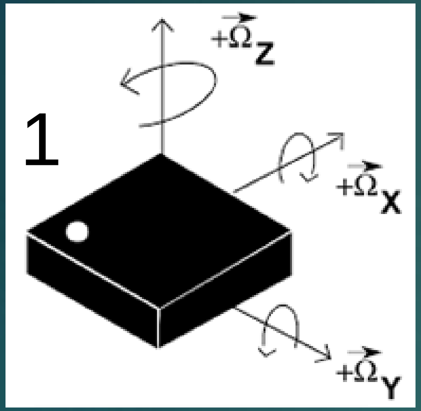
Off the Shelf Adapter PCB Converter Boards

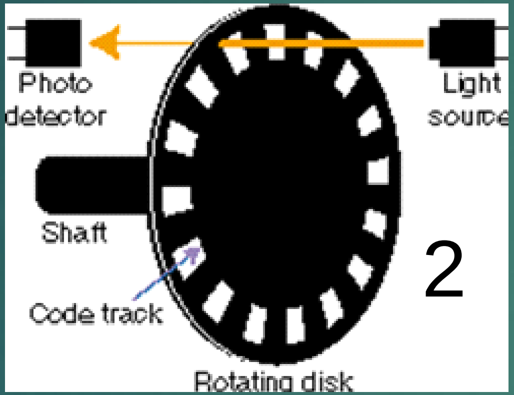


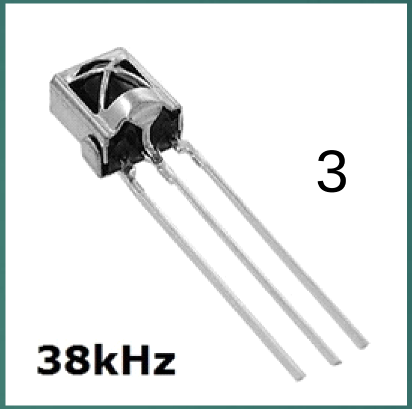
PCB Boards Designed for Specific MCU

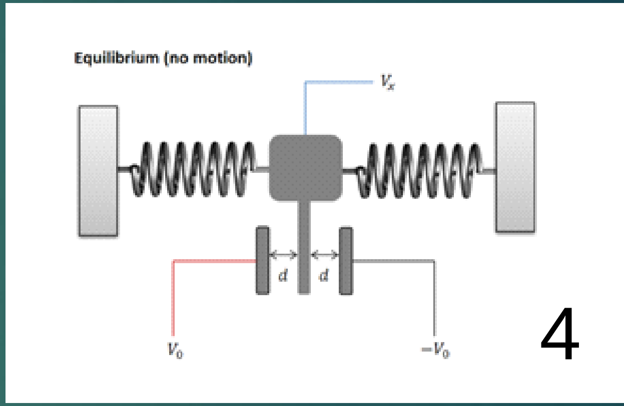


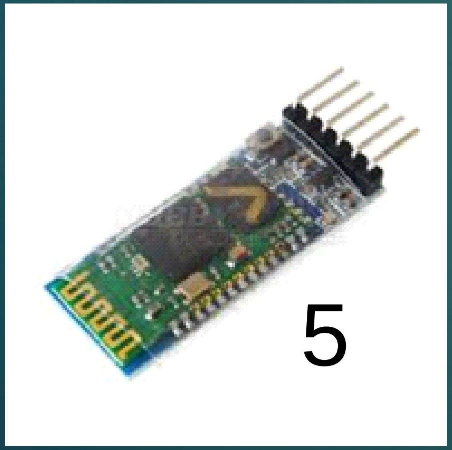
Name That Sensor/Module

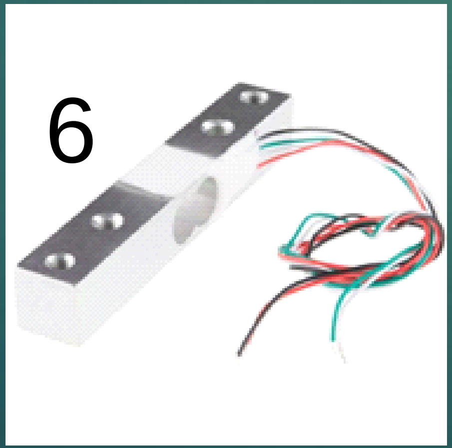


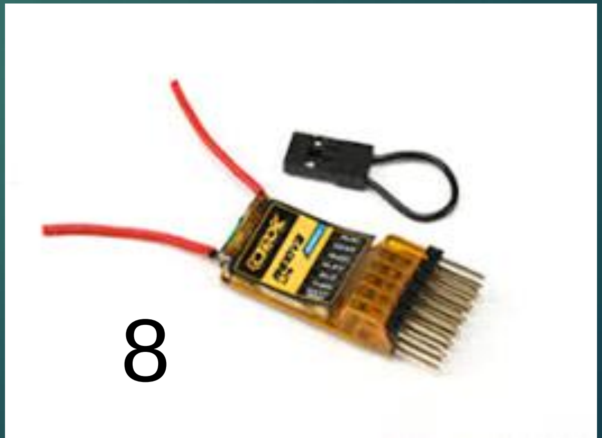












INTERFACING LEVEL & COMMUNICATION DEVICES

LEVEL DEVICES (Digital ON/OFF)

- INPUT(s)
  - One or more digital inputs hardwired to pins
  - Can be switch closures or pulses random or otherwise
- PULSES
  - Carry no other information other than the occurrence of an event
  - Pulse Width (or Pulse Position) contains coded information - in RC (Radio Control) pulse width contains data to position an RC servo motor

Processing Events

POLLING

- Pin(s) are continuously read until a change of state takes place
- Useful to initiate a start up
- Not very useful when other tasks need to be done

INTERRUPS

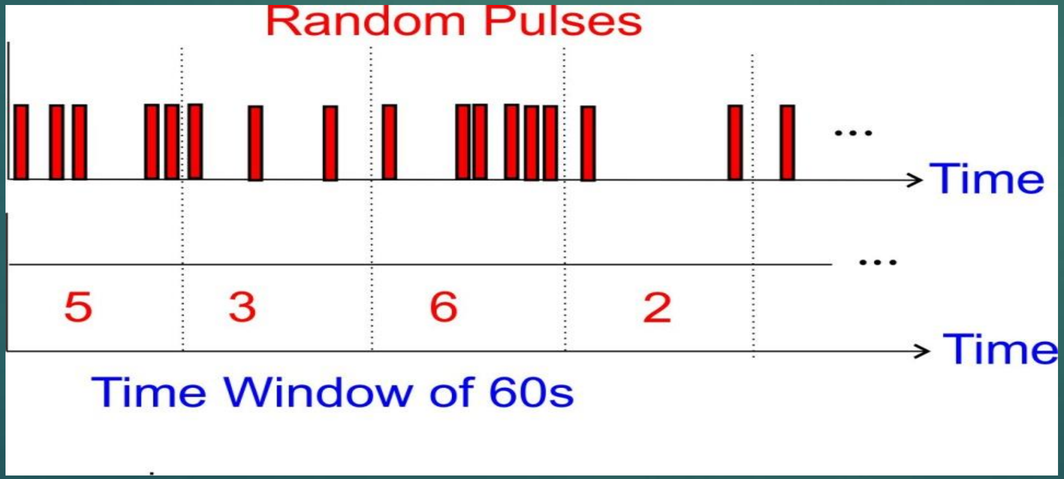
- An interrupt occurs when a change of state occurs in a hardwired pin
- The CPU saves its current state and immediately services the interrupt
- MCUs have many internal/external interrupts and are serviced according to priority

Interfacing Random Pulses From Radiation Detector

- Project Background
  - 207 Pachube IoT
  - Nuclear accident in Japan 2011 – Xively
  - Pachube -> LogMeIn –Cosm >Xively
  - 2013 Xively Public Cloud for the IoT
  - 2018 Xively purchased by Google

Interfacing Random Pulses From Radiation Detector

- THEORY
- The measurement of ionizing **radiation** is sometimes expressed as being a rate of **counts** per **unit** time. For low level of ionizing radiation, it is convenient to use **counts** per minute (CPM).
- Pulses from Radiation Detector are random ranging 0 CPM to many CPM



Interfacing Random Pulses From Radiation Detector

- PROGRAM DESCRIPTION
- INPUT
  - One hardwired pin configured to generate an interrupt on each leading edge of the random pulses
  - On Pin Interrupt - Count variable is increased by 1 - Interrupt is reset
- TIME WINDOW
  - Generated by a PWM (pulse width modulator)
  - PWM runs continuously independent of current code being executed by CPU
  - PWM generates a software interrupts at the end of each time window
    - Sets a Flag - Count is ready for processing - Software Interrupt is reset
- MAIN PROGRAM
  - Initializes variables
  - Loop on Flag (waits for a PWM to set Flag)
  - Process data
  - Display results
  - Back to Loop

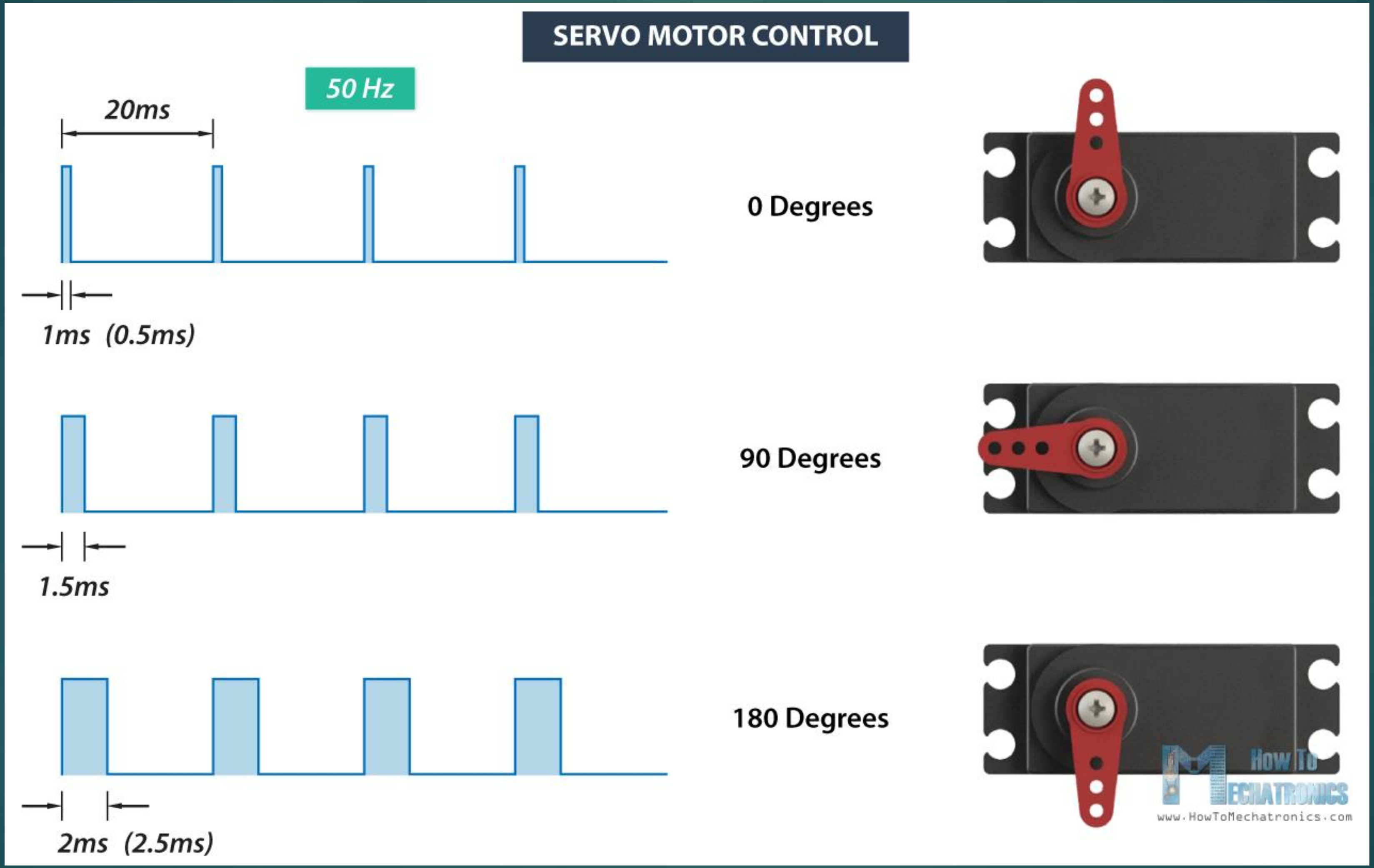
Interfacing Random Pulses From Radiation Detector

- Program Radiation Monitor
  - Using Cypress PSOC4

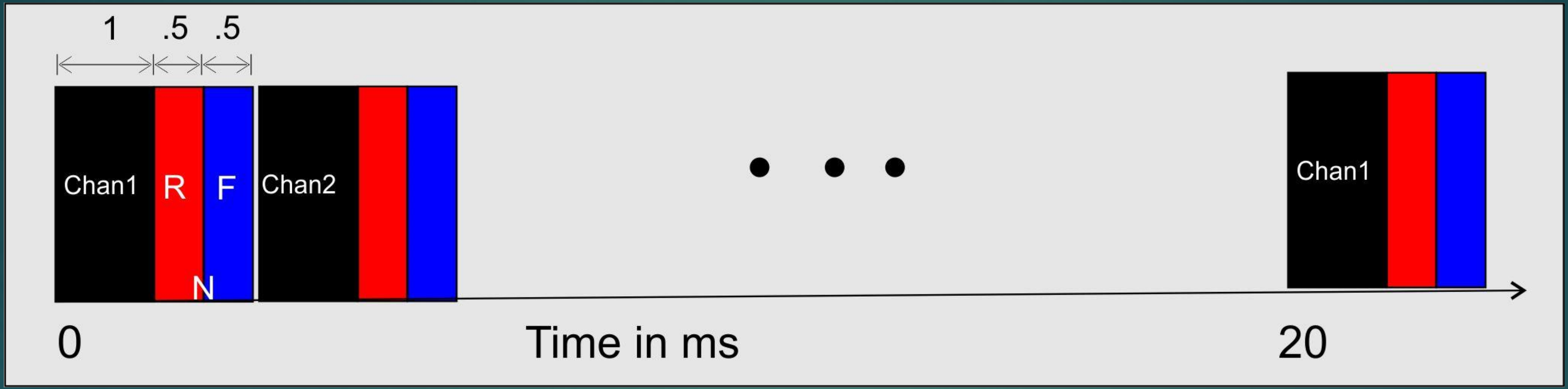
Interfacing Pulse Inputs with Coded Information

- ENCODING/DECODING PWM of RADIO CONTROLLED (RC) SERVO MOTORS
- RC Servo Motors have may used including
  - Radio controlled boats, planes, cars, robotics, cat/dog doors
- Are of special interest because they are easily controlled by MCUs without the need of a Radio TX
  - Come in all sizes, are inexpensive and can be modified internally for special applications
- A servo motor can be positioned by a MCU by suppling a PERIODIC PULSE in a specified time frame of 20 ms - 50 Hz
- The actual WIDTH of the PULSE (coding)determines the amount of rotation of a servo motor about a neutral axis.

Interfacing Pulse Inputs with Coded Information



Interfacing Pulse Inputs with Coded Information

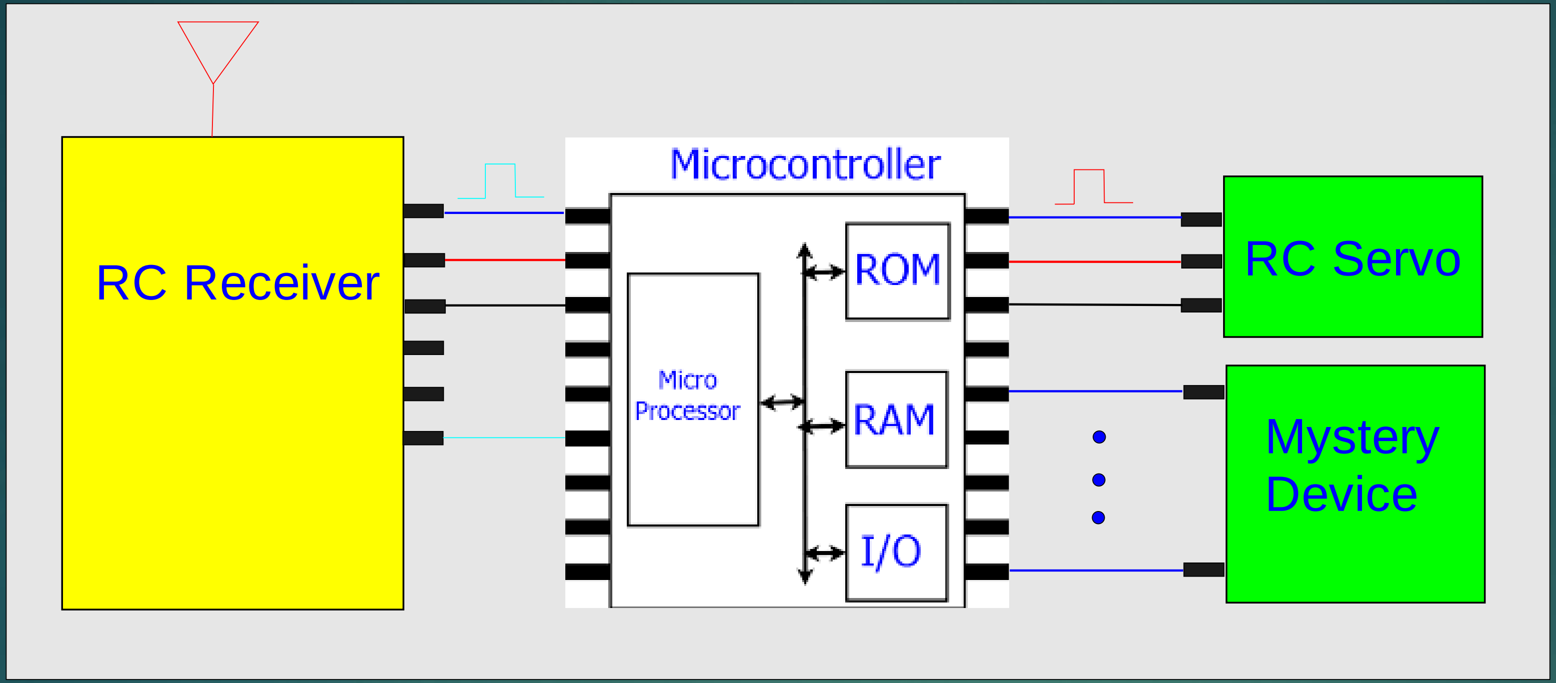


- The Pulse width to position a servo motor ranges from 1 MS to 2 MS or 5% to 10% of the period
- A convenient and flexible way is to use a 16-bit PWM
  - The Period of the PWM to 20 MS and not varied
  - The pulse width is then varied according to required position

Interfacing Pulse Inputs with Coded Information

- Demonstration Program
  - Uses a 16-bit PWM
  - At design time the period is set to 20 MS. The Pulse width set to 1.5 MS
  - On power up, the program waits for a switch closure
  - On each switch closure the servo motor cycles from extreme left, neutral, to extreme right corresponding to a rotation of -60° to 0° to $+60^\circ$

Decoding Radio Controlled Pulses from a Receiver

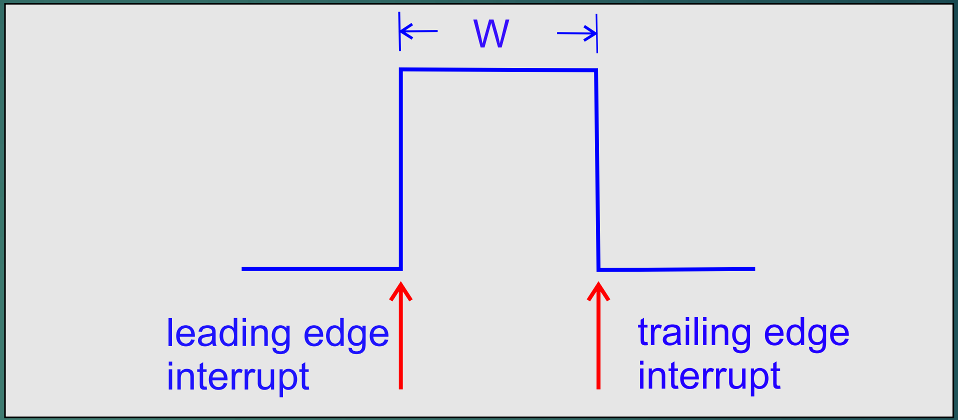


- Decoding the Pulse width has many interesting applications
  - The angular velocity of an RC Servo Motor can be reduced
  - The Mystery Device can be
    - A DC motor whose speed is proportional to the Pulse Width
    - A mechanical/electronic relay with OFF/ON function controlled by Pulse Width

Decoding Radio Controlled Pulses from a Receiver

Demo Program

Rate Reducer



- Input
  - Hardwired pin from receiver
  - Configured for Interrupts
- 16-bit Down Counter
  - Initialized and clocked to produce a count corresponding to a time in the range of 1 MS to 2 MS
- Output
  - Hardwired pin(s) to hardware device

INTERFACING LEVEL & COMMUNICATION DEVICES

Serial communication is either **Synchronous** or **Asynchronous**

- Synchronous serial communication uses a clock
  - 4-wire – SPI - Motorola
  - 3-wire – SPI- Maxim IC
  - 2-wire – I2C – Phillips Semiconductor
  - 1-wire – Dallas Semiconductor
  - 2- wire specific - Avia Semiconductor -HX711
  - 1-wire analog bus -DTMF

INTERFACING LEVEL & COMMUNICATION DEVICES

Asynchronous communication does not a clock

- Communication needs to be set to one of the standard communication rates (baud rate)
- Baud rates range from 110 to 25600 bits/sec with tolerance deviation of approximately 6%
- RS-232 – 2-wire unbalanced & referenced to ground
- RS-485 – 2-wire differential pair signals that improve noise immunity and distance

Interfacing Bluetooth SPP

- Bluetooth SPP
  - Emulates a serial cable to provide a simple substitute for existing RS-232, communication including the familiar control signals
  - "A serial cable is replaced by a secure wireless connection"
  - SPP Bluetooth Transceiver Modules are designed to connect to MCUs using RS-232 communication (UART Tx/Rx of an MCU)
  - Bluetooth Transceiver Modules can connect with each other or to PC, Cell Phones and Tablets
  - For Android devices download and install "**Bluetooth spp Pro**". It's free.

Interfacing Bluetooth SPP

Hardware Requirements

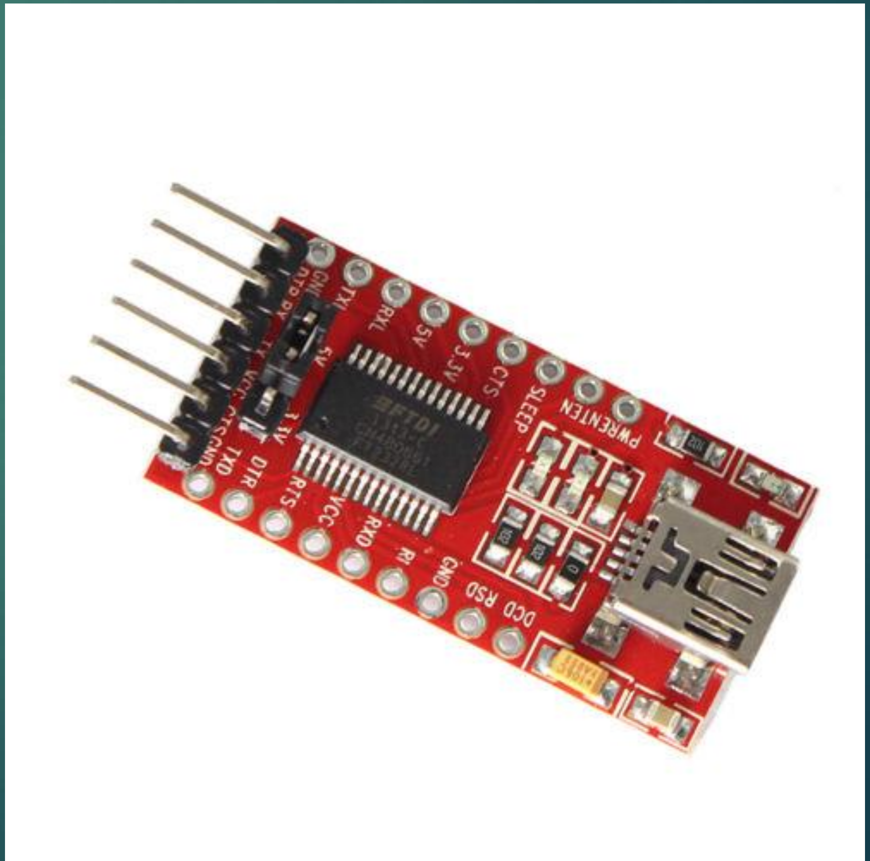
- One SPP Module (for cell phone/tablet) or two SPP Modules (using 2 MCUs)
  - Modules come in many forms
    - Some are transceivers (Tx & Rx)
    - Some are individual Tx or Rx
    - Some have fixed BAUD rates
    - The HC-05 is transceiver with baud rates up to 115200

- USB to Serial Converter Module
  - Modules plug into a USB port of a PC
  - The output of the modules are the RS-232 pins
  - Example- **Mini FT232RL 3.3V 5.5V FTDI**
    - Pins: DTR, RXD, TX, VCC, CTS, GND

Interfacing Bluetooth SPP

- HC-05 Bluetooth Module

FTDI USB to Serial Converter



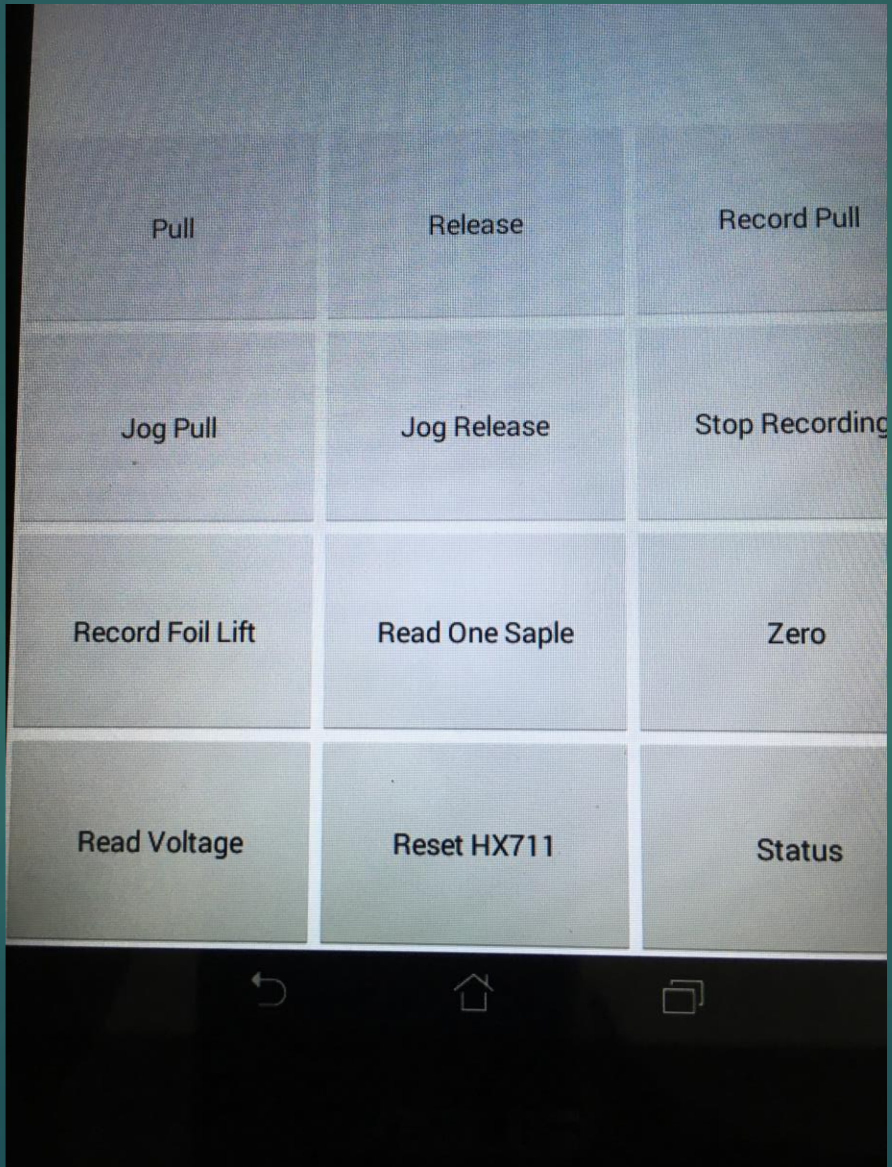
Interfacing Bluetooth SPP

- The HC-05 Bluetooth Module
  - HC-05 Bluetooth Modules are NOT ALL the same
  - Pin names and order may be different
  - Some have a binding switch
- Default Settings of HC-05
  - Baud Rate: 1200, Data Word:8 bits, Parity: none, Stop Bit: 1
  - PW 1234
  - Slave
- Changing Default Settings
  - Done by AT Commands
- Demo
  - Viewing and changing settings of a HC-05 SPP Module

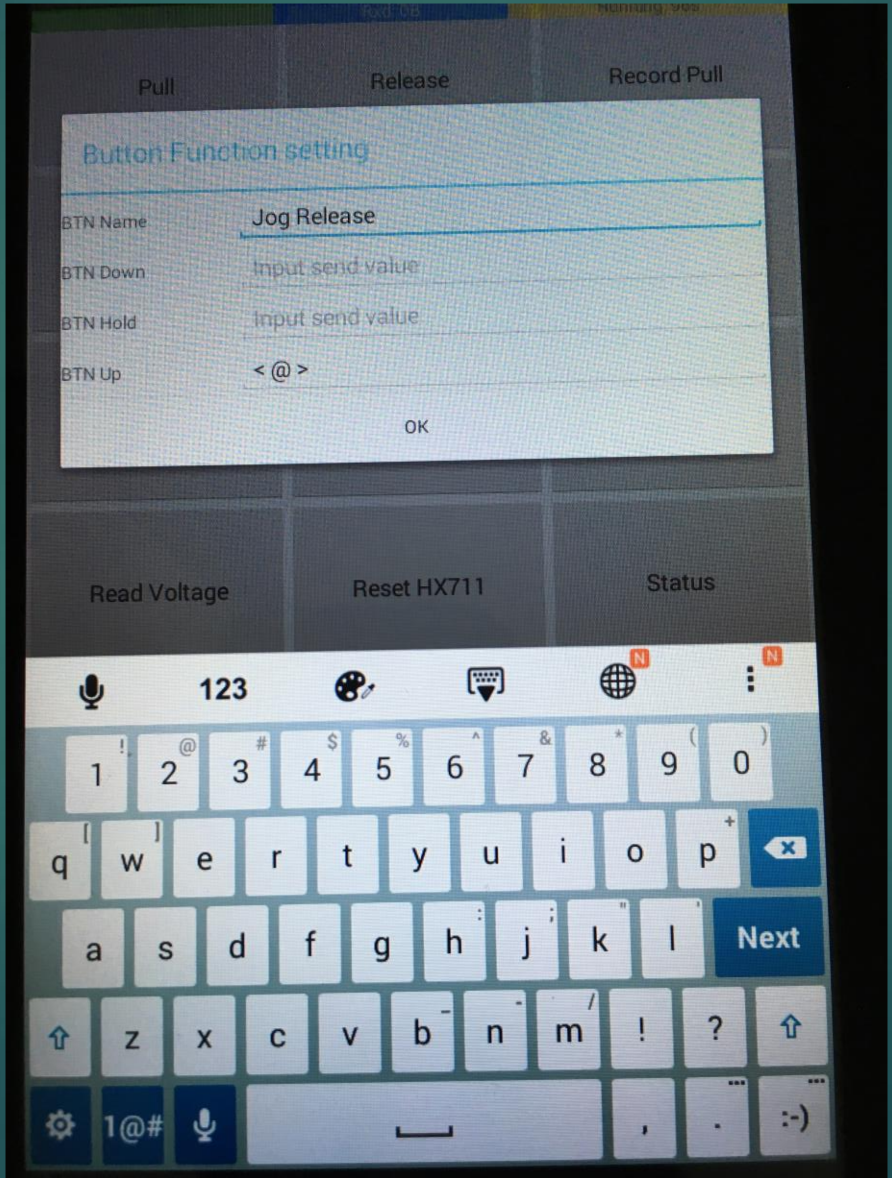
Interfacing Bluetooth SPP

- Demos
  - Viewing and changing settings of a HC-05 SPP Module
  - Cell Phone/Tablet

Interfacing Bluetooth SPP – Cell phone/Tablet



Interfacing Bluetooth SPP – Cell phone/Tablet



INTERFACING MOTION SENSING DEVICES BASED ON MEMS TECHNOLOGY

- MEMNS Micro-Electro-Mechanical Systems
  - Devices and structures that are made using the techniques of microfabrication
- MEMS SENSING DEVICES INCLUDE
  - Accelerometers, Gyroscopes, Magnetometers, Pressure
  - Combination of two or more of the above
- Examples
  - MMA7455 Tri-Axial accelerometer (3 DOF)
  - MPU-6050 (6 DOF) accelerometer & gyroscope
  - LSM9DSO -Adafruit accelerometer, gyroscope magnetometer
  - ADIS16480 (Analog Devices) -Ten DOF

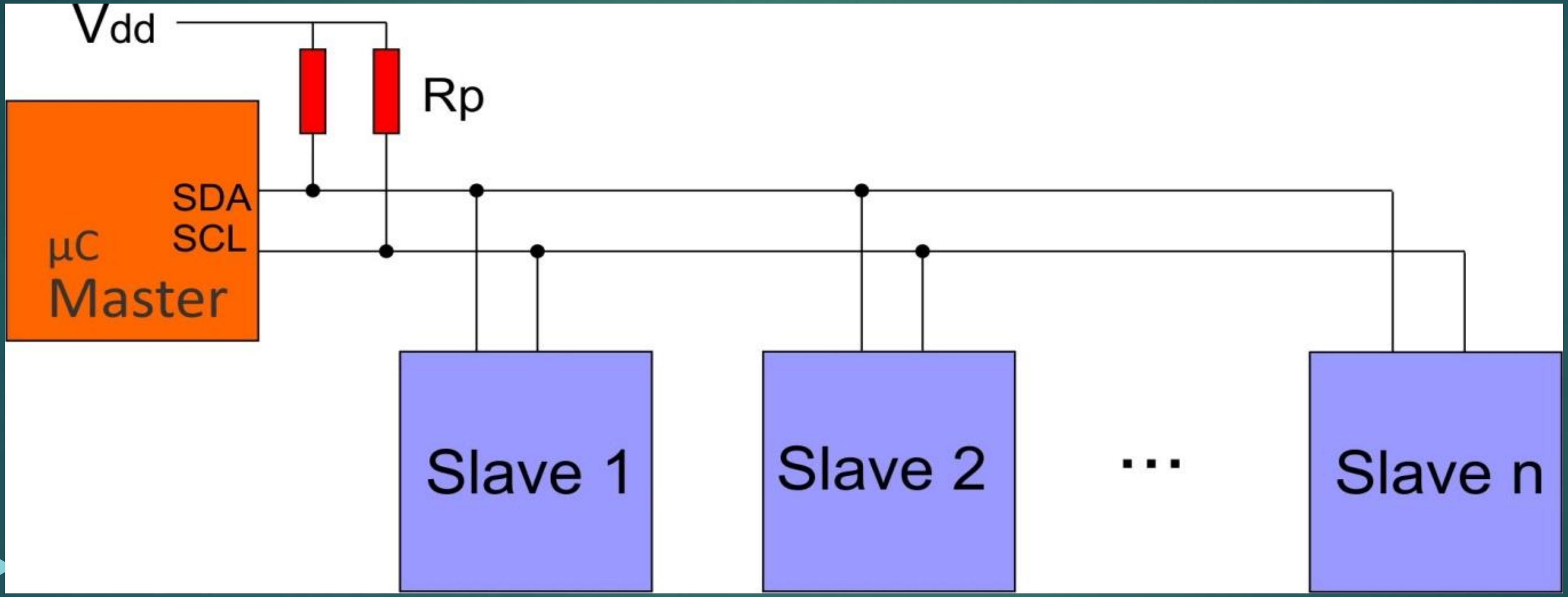
INTERFACING MOTION SENSING DEVICES BASED ON MEMS TECHNOLOGY

- COMMON FEATURES
  - Supply voltage 5 v or 3.0 V to 3.6 V
  - I/O pins 3.0 V to 3.6 V max
- Communication is Synchronous
  - I^2C, SPI, usually both

Interfacing MMA7455 Tri-Axial accelerometer Using I^2C Communication

- I^2C Communication
  - Popular because of its simplicity
  - More software overhead
  - Uses two signal wires with Pull Up resistors for communication
  - Communication speed can be 100 KHz or 400 KHz
  - There can be only one Master and many Slave
  - Master initiates ALL communication

Interfacing MMA7455 Tri-Axial accelerometer Using I^2C Communication

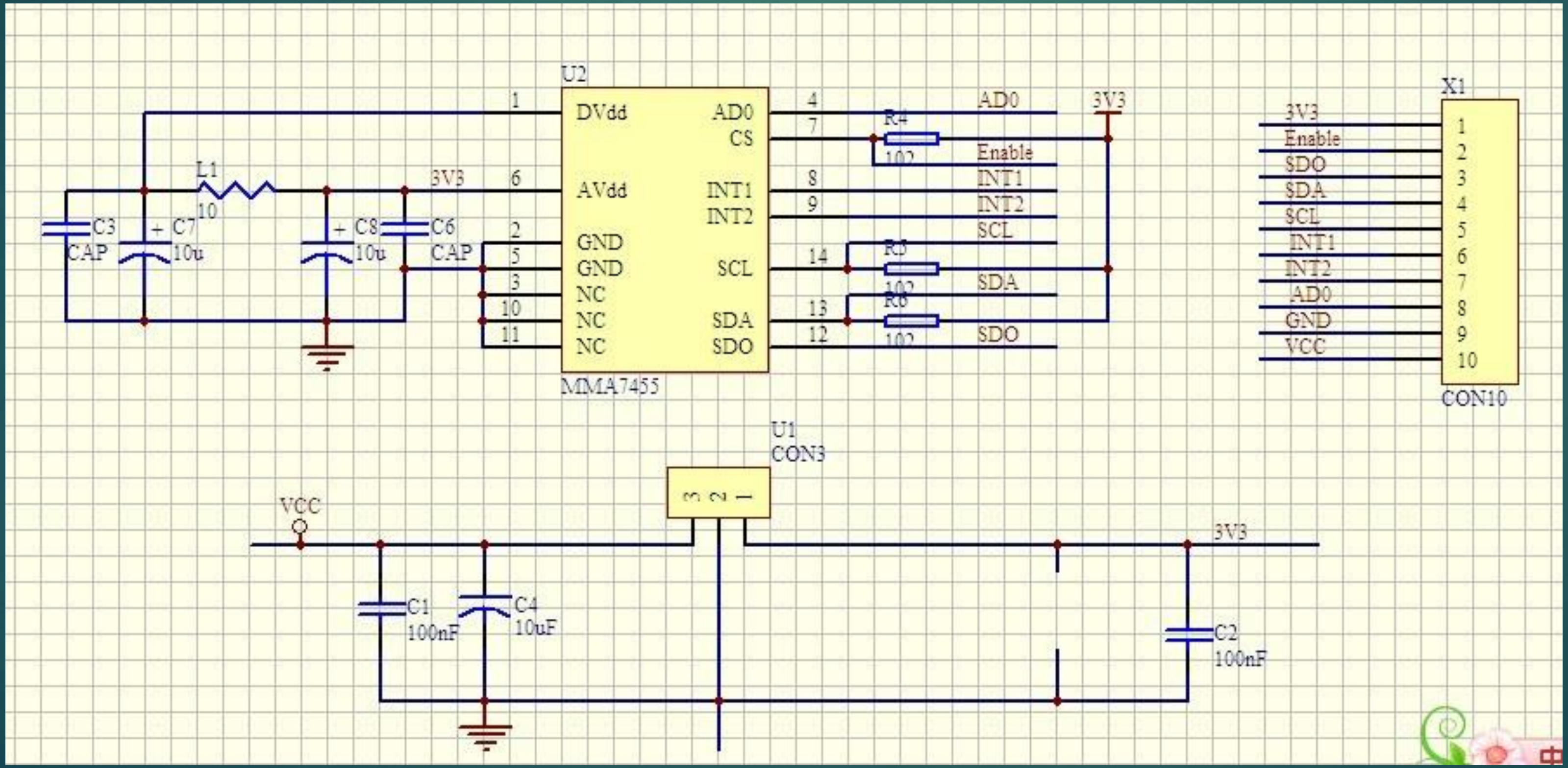


Interfacing MMA7455 Tri-Axial accelerometer Using I²C Communication

- MMA7455 Module

Interfacing MMA7455 Tri-Axial accelerometer Using I^2C Communication

- MMA7455 Module Schematics



Interfacing MMA7455 Tri-Axial accelerometer Using I^2C Communication

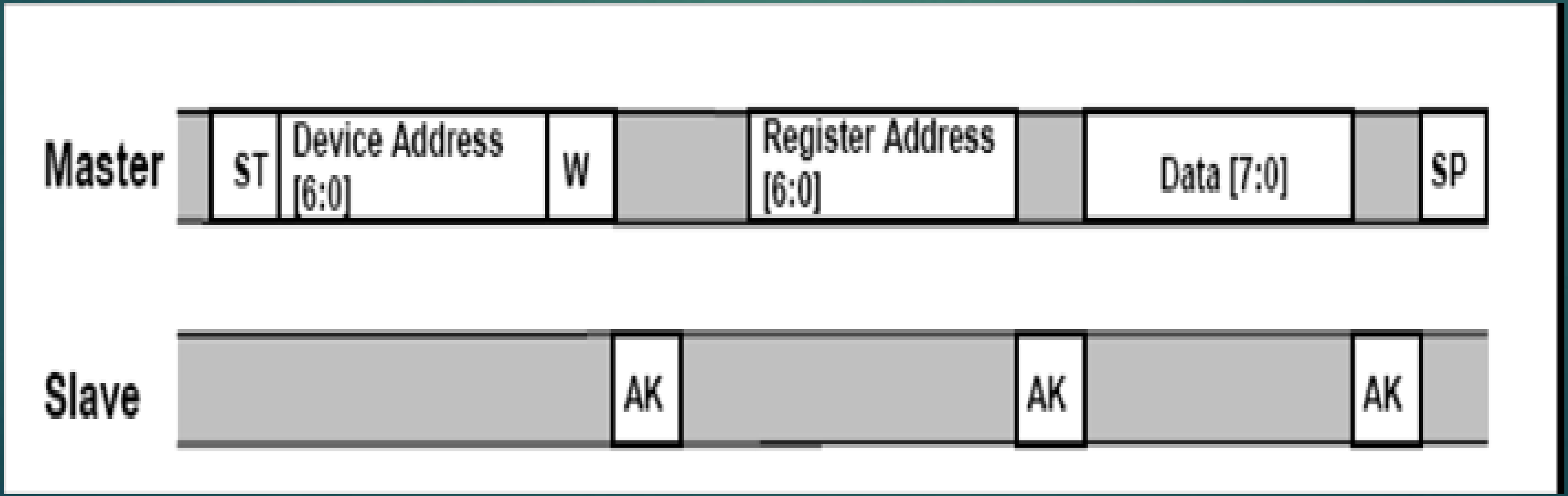
- Features
  - Digital Output (I2C/SPI)
  - Low Current Consumption: 400 µA
  - Self-Test for Z-Axis
  - Low Voltage Operation: 2.4 V – 3.6 V
  - Level Detection for Motion Recognition (Shock, Vibration, Freefall)
  - Pulse Detection for Single or Double Pulse Recognition
  - Selectable Sensitivity (±2g, ±4g, ±8g) for 8-bit Mode

Interfacing MMA7455 Tri-Axial accelerometer Using I^2C Communication

- Communication Protocol
  - All communication from and to the Master is in packets of 8 bits
  - Communication is initiated by the Master by sending a Start to the Slave using a 7 bit address Plus a R/W bit (0 /1)
  - R/W bit indicates whether to Write to or Read from the Slave
  - Slave acknowledges by sending AK
  - Communication continuous with Master sending the address of the register to write or read from followed by data
  - Communication is terminated by the Master sending a Stop Condition

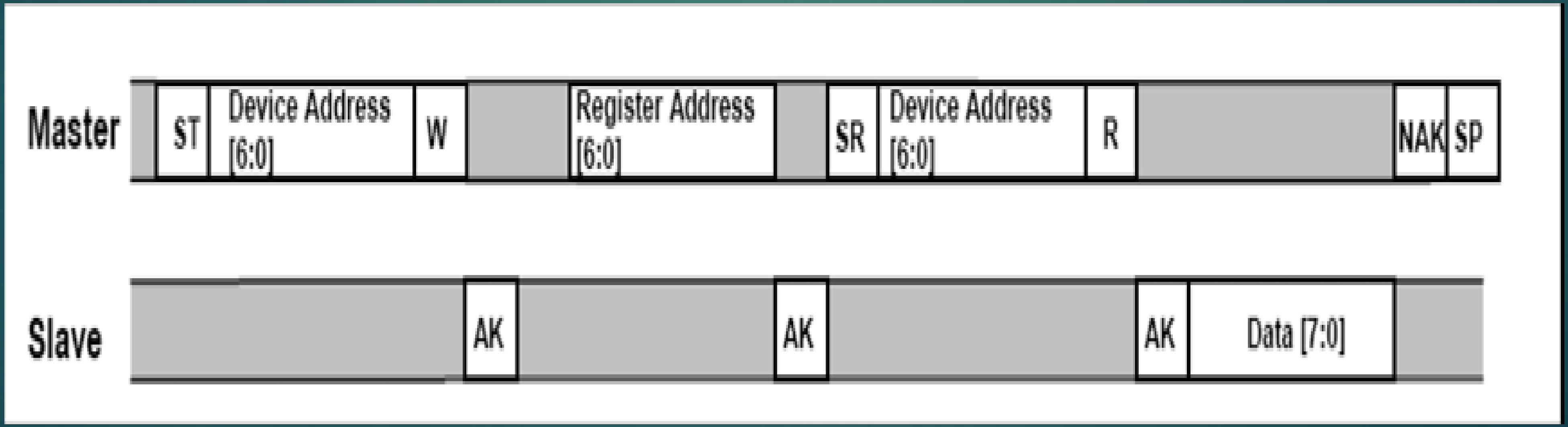
Interfacing MMA7455 Tri-Axial accelerometer Using I^2C Communication

- Example: To set Sensitivity ±2g
- Master Writing to a Single of the MMA7455L (R/W bit MSB)



Interfacing MMA7455 Tri-Axial accelerometer Using I^2C Communication

- Example: To read the acceleration on the x-axis
- Master Reading from a Single Register of the MMA7455L (R/W bit MSB)



Wing Control Actuating System

- Catamarans Flip!

Wing Control Actuating System

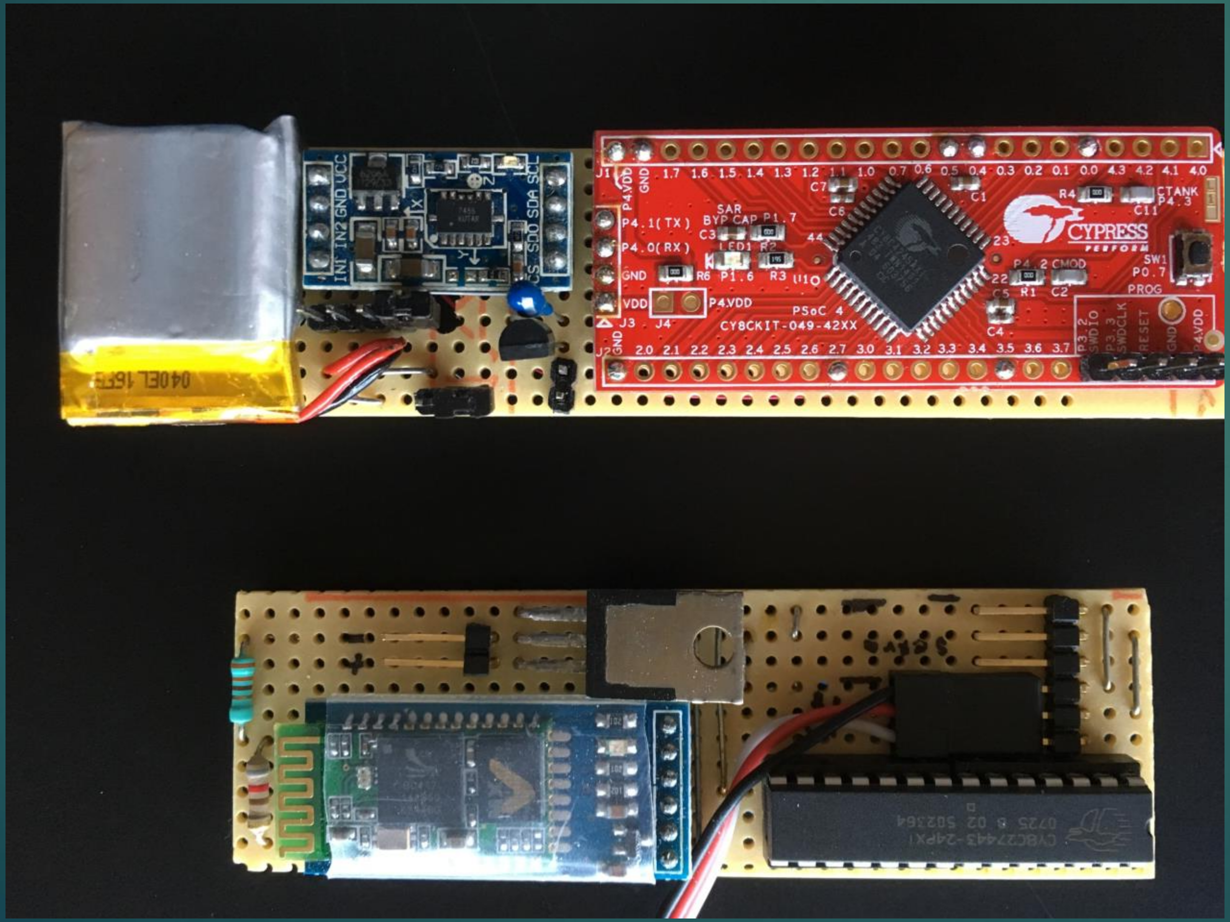
- Project Uses Two Separate Controllers Linked by Bluetooth
- Master Controller
  - Reads X.Y,Z Acceleration at a regular timed interval
  - Transmits to the Slave Module one of two control codes based on pre-set limits on each of the three axis
  - Control codes indicate normal condition- do nothing or remove RC from operator

Wing Control Actuating System

- Slave Controller
  - Continuously reads Control Codes from Master Controller
  - Under Go conditions the RC signal from receiver is decoded by the MCU and passed on to the servo controlling the tail. Operator has full RC control
  - Under No Go the MCU RC signal is replaced by a 1 ms pulse to put the tail in Neutral position

Wing Control Actuating System

- Controller Used In This Project



Wing Control Actuating System

- Program
- Demo

END