Arduino Day 2017

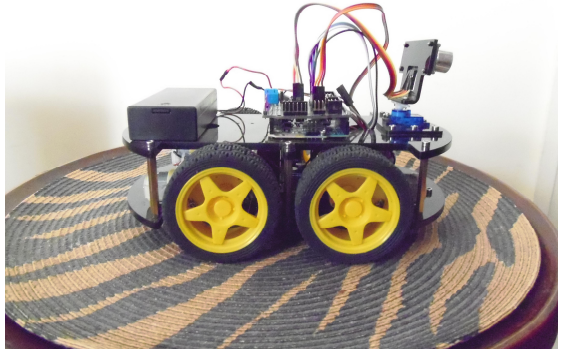
Arduino for Real-Time Control

Objectives

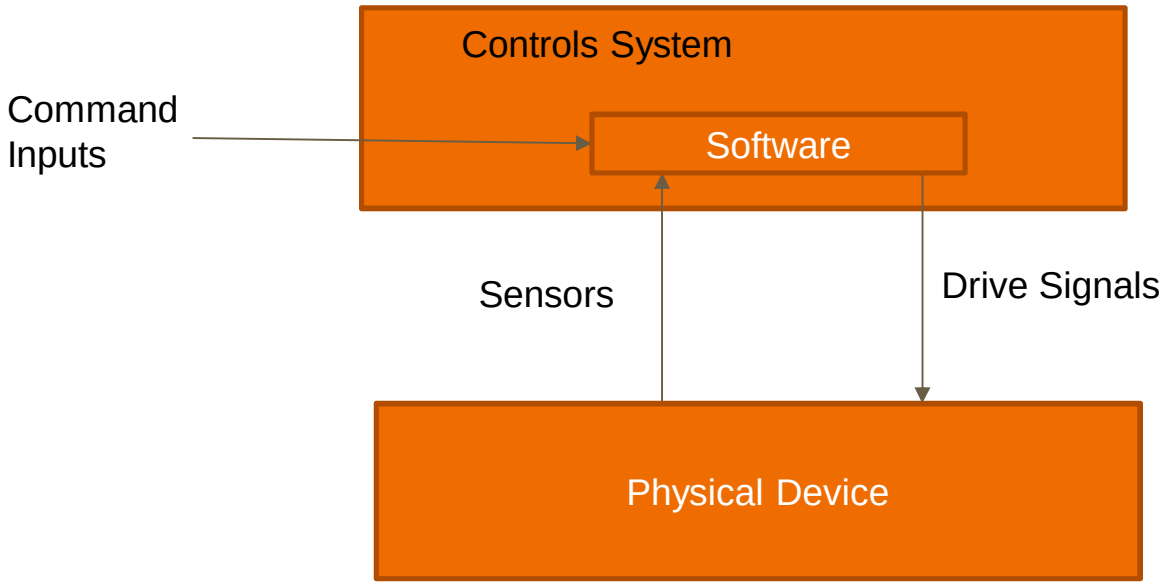
- Example of Real-Time System
- Example of Real-Time System Software Design
- Some Techniques to get you started.
  - Fixed Time Schedule
  - Designing and Implementing State Machine
  - BLE Example of a State Machine
  - Collision Avoidance Interrupt

What is Real-Time Control System?

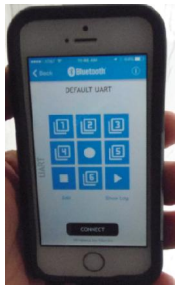
A microprocessor based system that is paired with electrical/mechanical device that produces the desired performance by compensating for environment and the physical device limitations.



Common to all Real-Time Control Systems



Robot Car + Arduino 101



Bluetooth LE

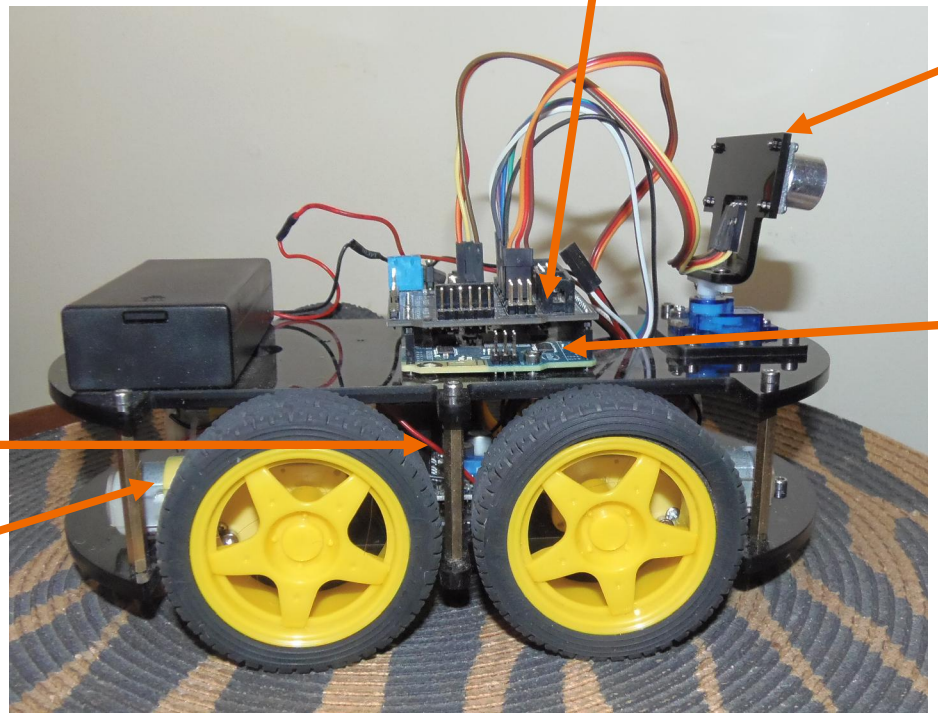
Arduino Sensor Shield V5.0

Ultrasonic Sensor and Servo

Arduino 101

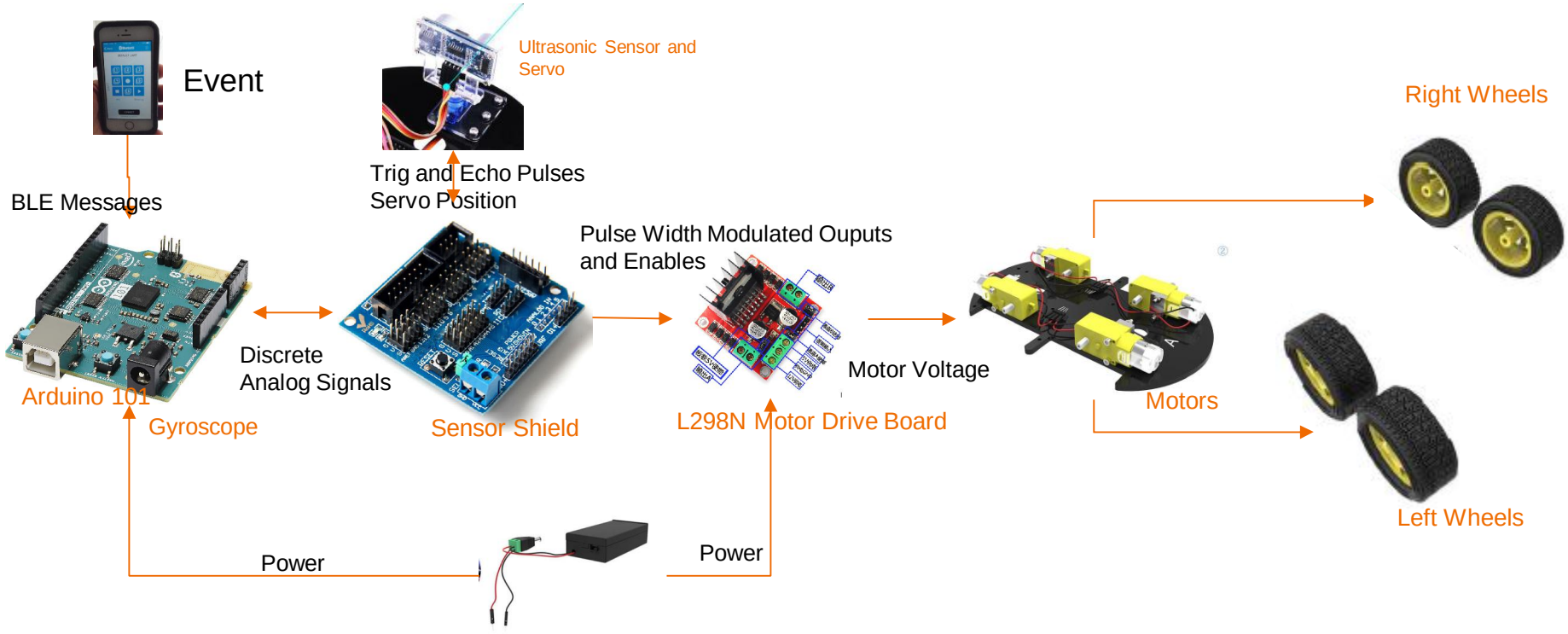
L298N Motor Drive Board

4 Motors



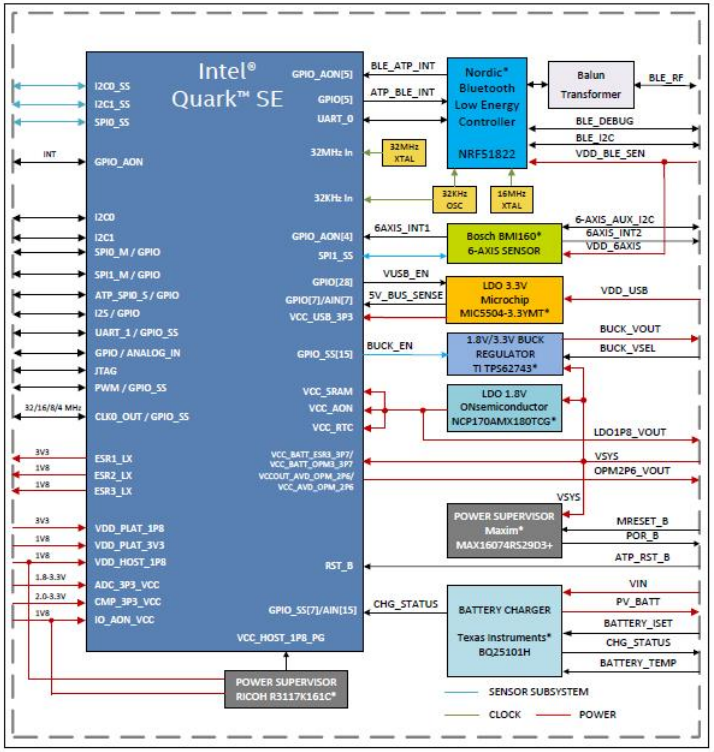
Elegoo Robot Car - UNO + Arduino 101

Hardware Block Diagram



Features Intel Curio of the Arduino 101

Figure 1-1. Intel® Curie™ module block diagram

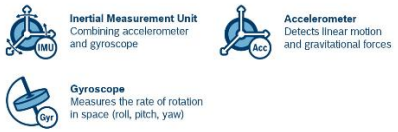


Intel Quark microcontroller
32 bit – Pentium x86 core.

384kb of flash
80kb of SRAM

ARC EM-4 based Sensor Subsytem

Six-axis accelerometer/gyroscope (Inertial Measurement Unit)



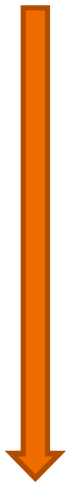
Bluetooth Low Energy

Pattern Matching Engine (Neural Net)

USB, I2C,I2S, UART, SPI, DMA Controller, GPIO, PWM
ADC Unit, Analog Compartors, RTC

Real Time Systems Cannot Wait to Think

Time in Seconds



Check for Event

Process the Event

Go Forward

Check for possible Collision

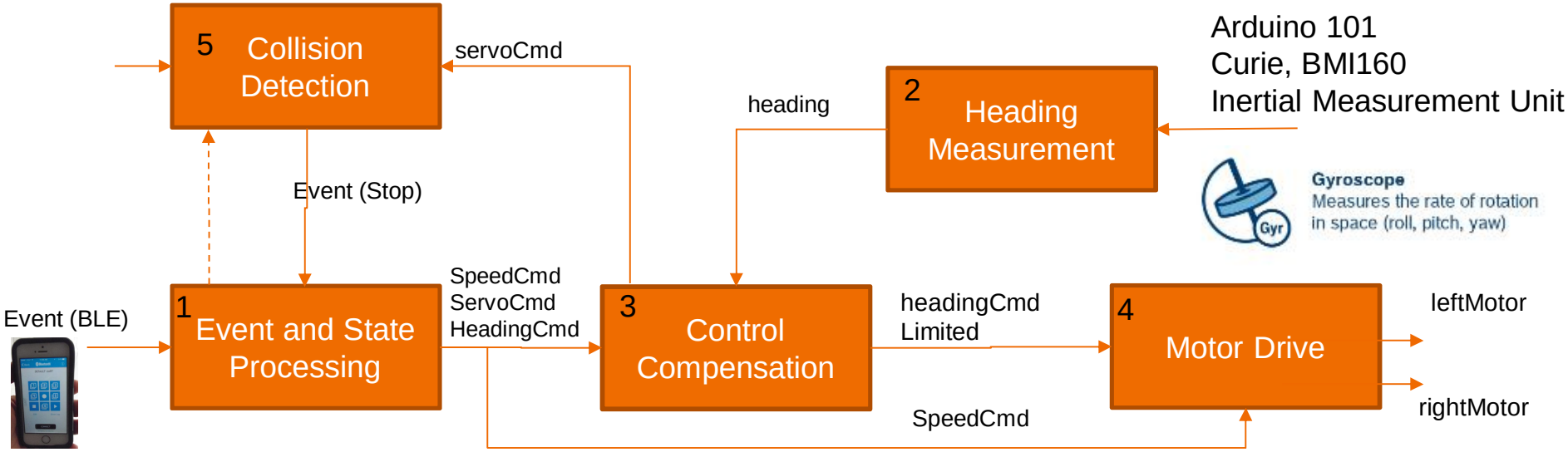
Check for Event

Go forward

Just would not fly…….

Demonstration – Event and State

Software needs to take action in “Real-Time”



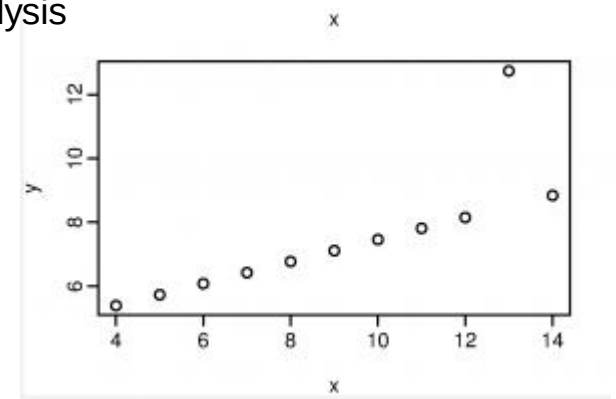
Perform 25 times a second or every 40 millisecond.

1,2,3,4,5 1,2,3,4,5 1,2.3,4,5 1,2,3,4,5 ……

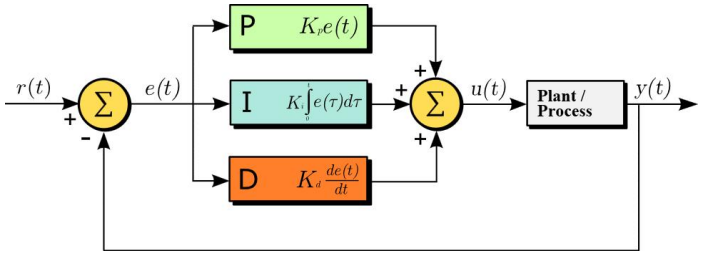
Real-Time
Execution needs Fixed Interval Execution

In order to use advanced control methods you should always design your system to sample its signals and execute algorithms at a high fixed rate. At least 10 times a second.

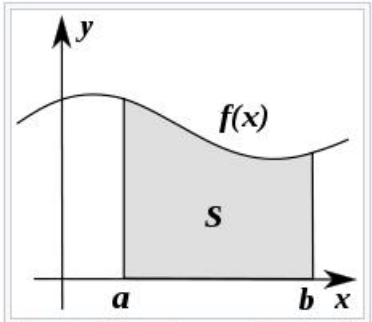
Data Analysis



PID Controller



Numerical Integration



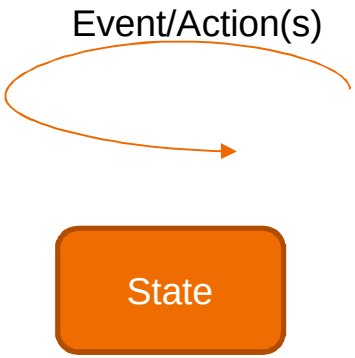
Numerical integration consists of finding numerical approximations for the value S

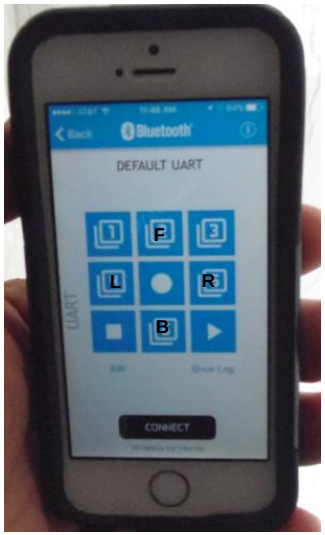
Code for Setting up a Fixed Intervals Execution

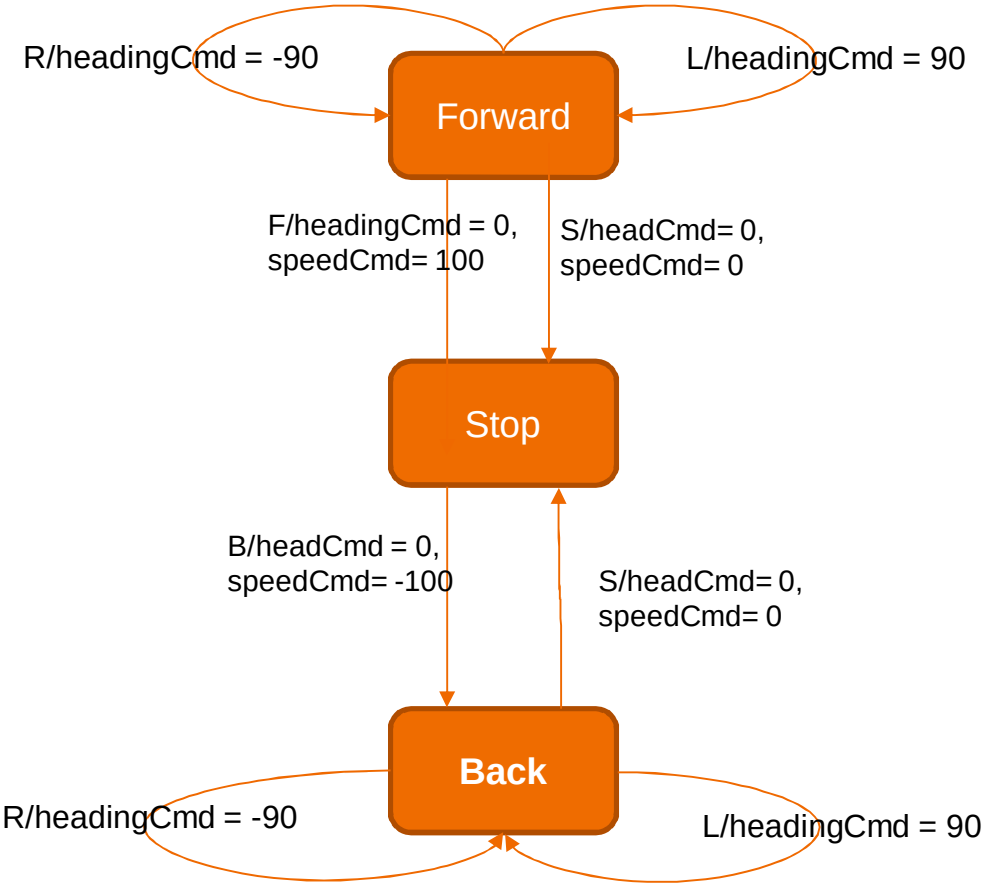
```
void setup() {
// initialize variables to pace updates to correct rate
   microsPerReading = 1000000 / 25; // .04 seconds or a 25Hz sample rate.
   microsPrevious = micros()
   // Insert additional setup code
}

void loop() {
 // check if it's time to run
    microsNow = micros();
    if (microsNow - microsPrevious >= microsPerReading) {


    // Insert you fixed time code that will execute 25 times a second.


     // increment previous time, so we keep proper pace
     microsPrevious = microsPrevious + microsPerReading;
    } else {  Serial.print("Timing not met!");  };
  // other code not fixed time can go here....
}
```

Event and State processing is used to support real time decisions

Event state processing is a typical technique in real time system. Events and actions are taken based on your systems state.

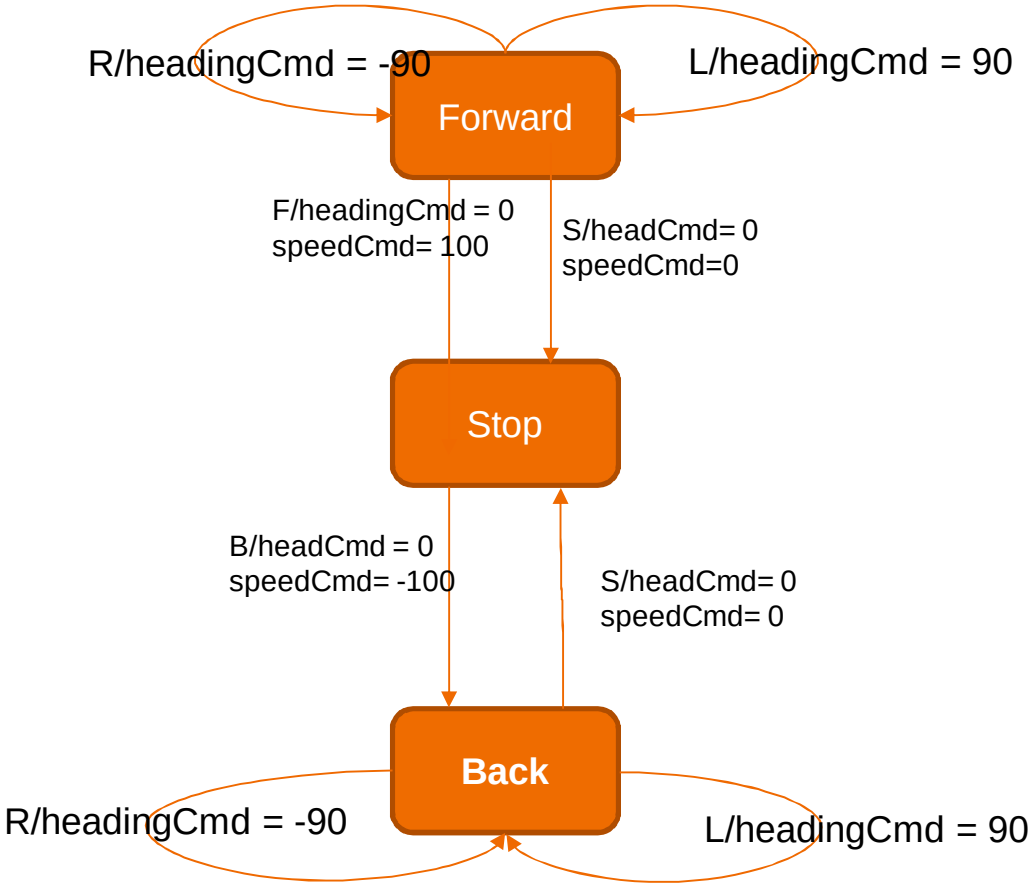






Mealy Diagram

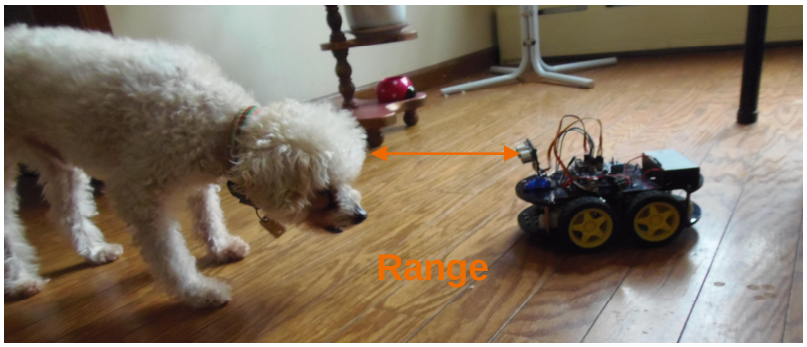
Code For State Machine

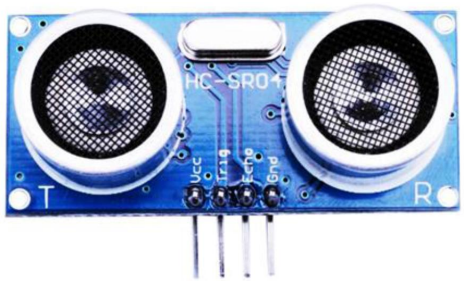


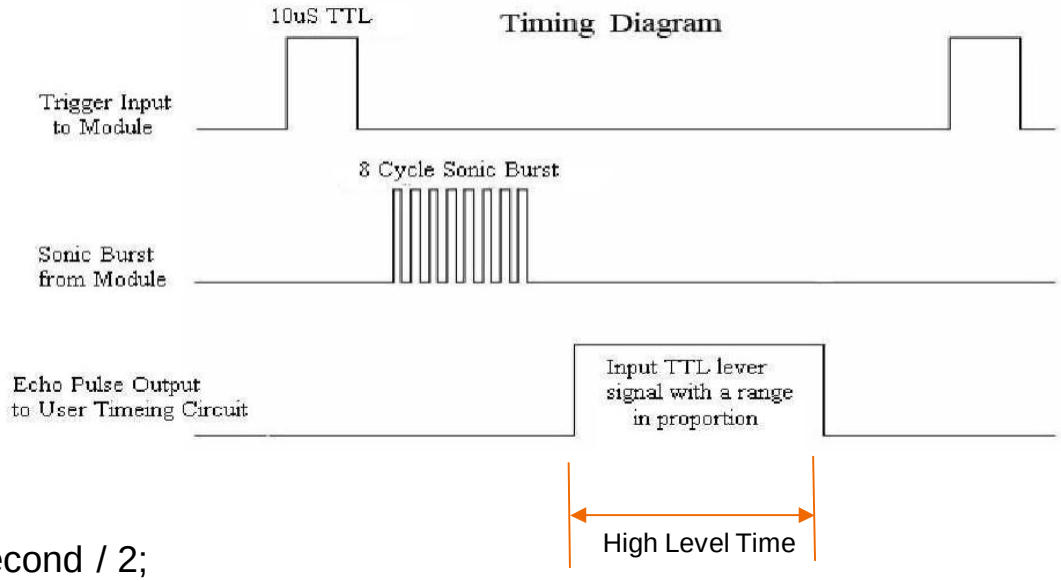

```
if (event == F)
{  headingCmd =  0;
   speedCmd = 100;
   state = “Forward”;

} else if( event == R  )
{   headingCmd = -90;

} else if (event == L )
{   headingCmd =  90;

} else if (event == B )
{  headingCmd =  0;
   speedCmd = -100;
   state = “Backward”;

 } else if (event == S )
 {  headingCmd =  0;
    speedCmd = 0;
    state = “Stop”;
  }
```

Collision Avoidance Processing

Collision Avoidance detects an object with in a certain minimum range and producing a stop event.





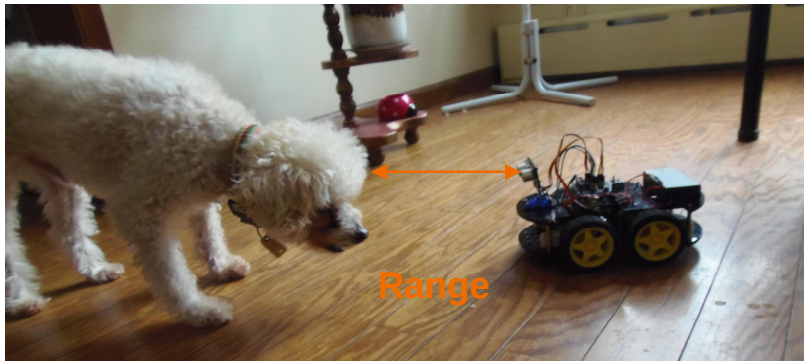


Range = High Level Time * 340 Meter/Second / 2;

(Speed of sound)

Demonstration Collision Avoidance

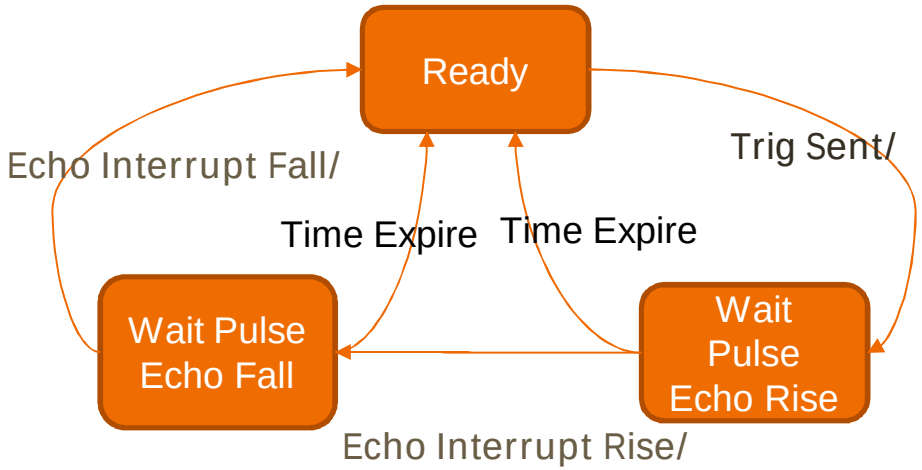
Live Demo



Collision Avoidance State Machine

The device has three states that occur during short distance processing.

However there is a short cut path when objects are further away.



Collision Avoidance Interrupt Code

```
int Echo = A4;
int Trig = A5;
volatile float Distance;
volatile int echopulseStart;
volatile int echopulseEnd = 0;
volatile int echopulseLength = 0;


int Distance_Measure_Start()
// This runs to kick off a measurement of Distance
{if ( obstAvoidState > 0 ) {
     Distance = 10000;
  }
  digitalWrite(Trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(20);
  digitalWrite(Trig, LOW);
  attachInterrupt(digitalPinToInterrupt(Echo), Distance_EchoPulse_Start, RISING);
  obstAvoidState = 1;
}
```

This function sends the trigger pulse to the Ultrasonic Unit. It will be called from loop code to kick off sample.

Collision Avoidance Interrupt Code

These two functions are executed by interrupts on the Echo Pin. This one when Pulse goes RISING. I get samples time when it occurs.

```
void  Distance_EchoPulse_Start()
//This detects the Echo Pulse Going High
{
  if ( obstAvoidState == 1 ) {
  echopulseStart = micros();
  attachInterrupt(digitalPinToInterrupt(Echo), Distance_Echo_Compute, FALLING);
  obstAvoidState = 2;
  }
}


void Distance_Echo_Compute()
{if ( obstAvoidState == 2 ) {
    echopulseEnd = micros();
    if (echopulseEnd > echopulseStart ) {
        echopulseLength = echopulseEnd- ec
        Distance =  echopulseLength /58;
    }
    obstAvoidState = 0;
  }
}
```

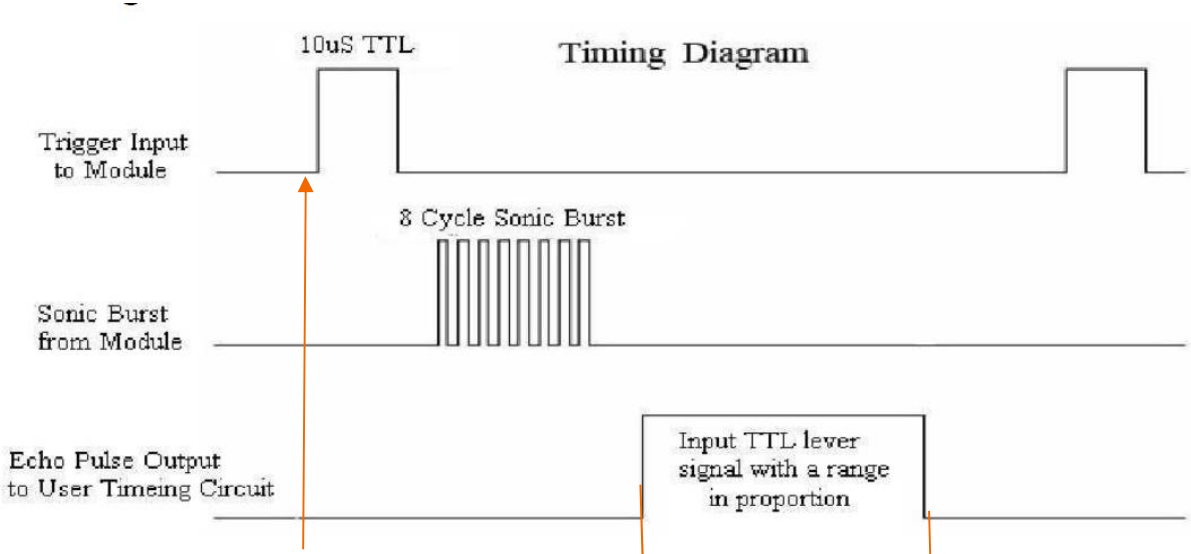
Execute when Pulse goes FALLING. It samples the time and computes the Distance.

Collision Avoidance Interrupt Code

```
//Setup Code
  pinMode(Echo, INPUT);
  pinMode(Trig, OUTPUT);


// Loop Code
    Distance_Measure_Start();
```

This code initiates the conversion every 40 milliseconds as per a fixed interval.

Collision Avoidance Interrupt Code




```
Distance_Measure_Start()
State=1;//Wait Echo Pulse Rise
```

```
Distance_EchoPulse_Start()
State=2;//Wait Echo Pulse Fall
```

```
Distance_Echo_Compute()
State=0;//Ready
```

Summary

- Example of Real-Time System
- Example of Real-Time System Software Design
- Some Techniques to get you started.
  - Fixed Time Schedule
  - Designing and Implementing State Machine
  - BLE Example of a State Machine
  - Collision Avoidance Interrupt

MAKE

GIVE

TOOL UP

SHARE

LEARN

Thank You

SUPPORT

PARTICIPATE

CHANGE

PLAY