

# Microcontroller Programming

# Robot Hardware

- Two main components to program
  - Motors – Give Output
  - Sensor Panel – Get Input

# Controlling Motors

- Uses Pulse Width Modulation
- L298 Driver IC
  - Can bear the load current of the motors
  - Can control direction of rotation
  - Has 3 inputs (for each motor)
    - One PWM pin
    - Two direction pins

# Sensors

- 8 IR sensors used
- OP-AMP circuit is used to convert analog signal to digital input
- Each sensor pin outputs 5V in white background and 0V in black background
- Software functions exists to check input state

# First Program – Blinking a LED

- Software Used – PIC C
  - Easy to use
  - Automatic configuration settings
  - No need of in-depth knowledge
- Creating a new Project with PIC C - Demo

# Blinking a LED - Demo

Steps:

1. Apply voltage to a Pin
2. Apply a delay
3. Remove voltage from Pin
4. Apply a delay
5. Go to Line 1

# Blinking a LED - Demo

Software Functions needed:

- Naming – PIC C names pins as PIN\_A0, PIN\_A1 etc.
- Apply Voltage Output - `output_high(PIN_NAME);`
- Remove Voltage - `output_low(PIN_NAME);`
- Apply a delay - `delay_ms(Time_In_Milliseconds);`
- Loop – `while(1){ Code_To_Loop }`

# Blinking a LED - Demo

Actual Code:

```
while(1){  
    output_high(PIN_A0);  
    delay_ms(1000);  
    output_low(PIN_A0);  
    delay_ms(1000);  
}
```



# Second Program – Reading Input

Steps:

1. Check Status sensor input
2. If input is high, light up a LED
3. Else, don't light up the LED
4. Go to line 1

# Second Program – Reading Input

Software Functions needed:

- All functions from previous example
- Read input status - `input (PIN_NAME)`

Usage:

```
Value = input (PIN_NAME);
```

Value is a variable.

If input is high, value = 1

If input is low, value = 0

# Second Program – Reading Input

Actual Code:

```
while(1){  
    int value;  
    value = input(PIN_B0);  
    if ( value == 1) { output_high(PIN_A0); }  
    else{ output_low(PIN_A0); }  
}
```

# Controlling Motors

Steps:

1. Setup PWM Configuration – Automatically done by PIC C software
2. Set motor directions
  - One direction pin – high state
  - Other direction pin – low state
3. Set rotation speed (PWM duty cycle)

# Controlling Motors

Our Configuration:

Left Motor: direction pins – RD6, RD7  
PWM pin – RC2/CCP1

Right Motor: direction pins – RD5, RD4  
PWM pin – RC1/CCP2

# Controlling Motors

- Demo – setting PWM configuration
- Software functions

`set_pwm1_duty(value)` – speed for left motor

`set_pwm2_duty(value)` - speed for right motor

Value is a integer variable between 0 and 1023. speed changes according to this value.

# Controlling Motors

Actual code for left motor:

```
output_high(PIN_D6);  
output_low(PIN_D7);  
set_pwm1_duty(100);  
while(1);
```