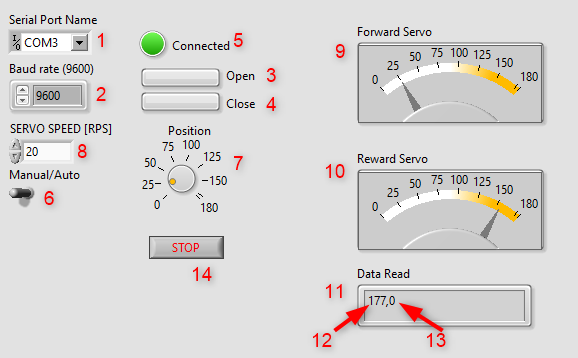
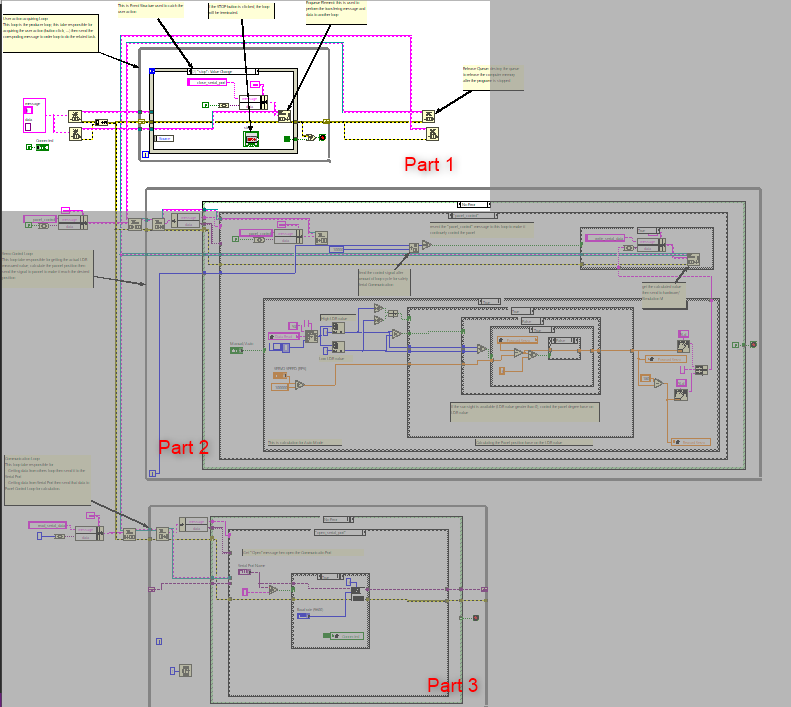
# User Interface for controlling the motor system



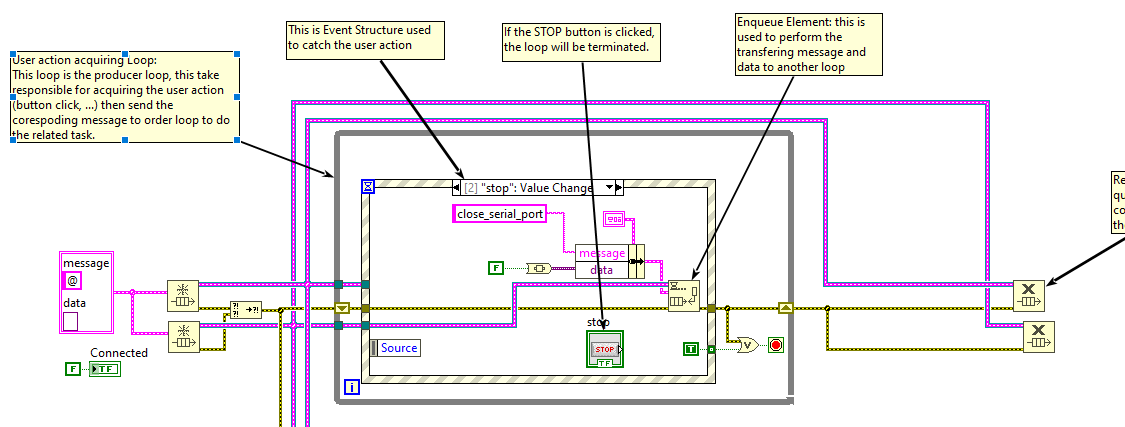
1. Serial Port Name:  
   The name of the Serial Port available for communication (to communicate to the hardware which control the motor or the software which simulate the motor).
2. Baud rate:  
   The baud rate of the Serial port, 9600 is mostly use due to its stable behavior.
3. Open button:  
   Click to open the port and start the communication.
4. Close button :  
   Click to close the port and stop the communication.
5. Connected:  
   The LED to indicate that the communication is ready to be used (send/receive data).
6. Manual/Auto Switch:  
   User can choose operating mode by toggle this switch.  
   In Manual mode, the servo position is get from the Position Knob value.  
   In Auto mode, the servo position is calculated based on the High LDR value and Low LDR value.
7. Position Knob:  
   Control the Servo position in Manual mode.
8. SERVO SPEED:  
   Indicating how fast the servo will turn.
9. Forward Servo:  
   Indicate the forward servo position.
10. Reward Servo:  
    Indicate the reward servo position.
11. Data Read:  
    The data read from Serial Port.
12. The value of High LDR read from Serial Port.
13. The value of Low LDR read from Serial Port.
14. Stop button:  
    Click this button to stop the program.

**Diagram:**



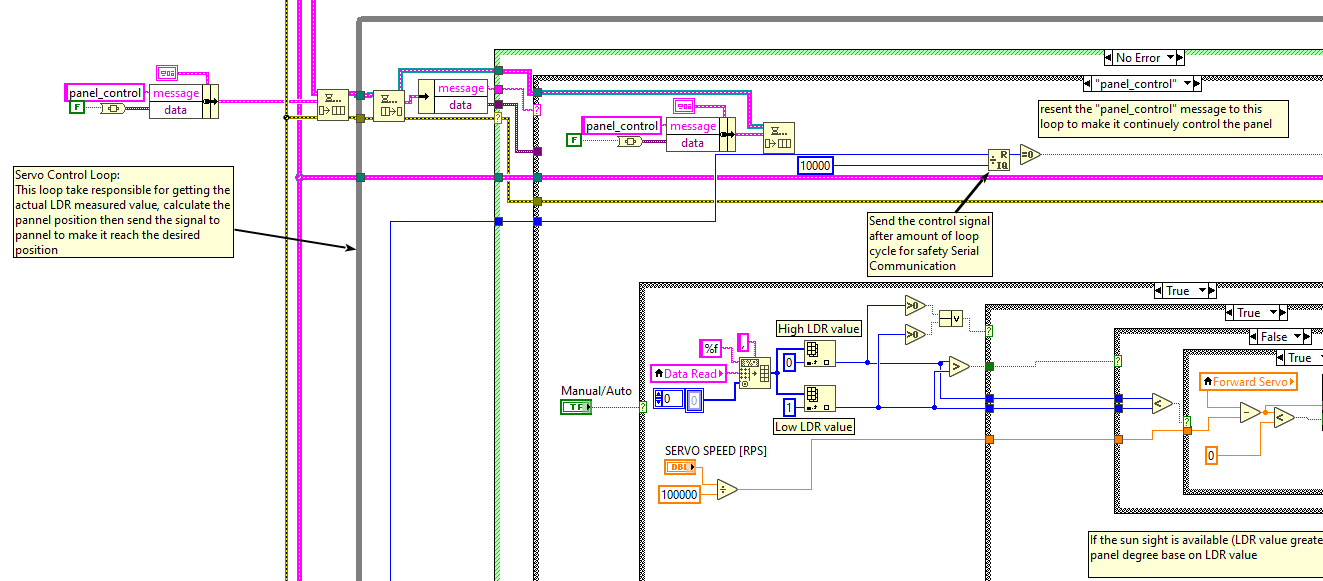
1. **Part 1:**

User action acquiring Loop:  
This loop is the producer loop, this take responsible for acquiring the user action (button click, ...) then send the corresponding message to order loop to do the related task.



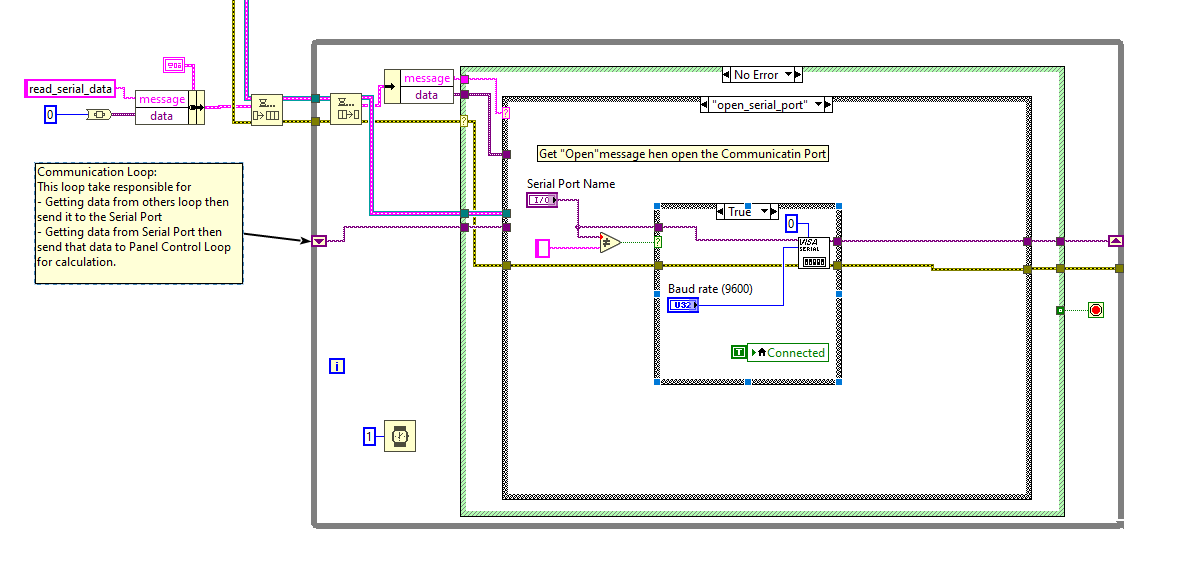
1. **Part 2:**

Servo Control Loop:  
This loop take responsible for getting the actual LDR measured value, calculate the pannel position then send the signal to pannel to make it reach the desired position

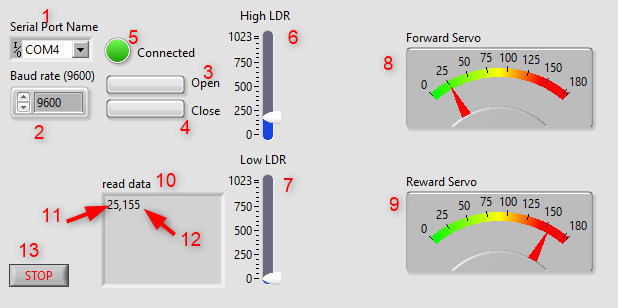


1. **Part 3:**

Communication Loop:  
This loop take responsible for   
- Getting data from others loop then send it to the Serial Port  
- Getting data from Serial Port then send that data to Panel Control Loop for calculation.

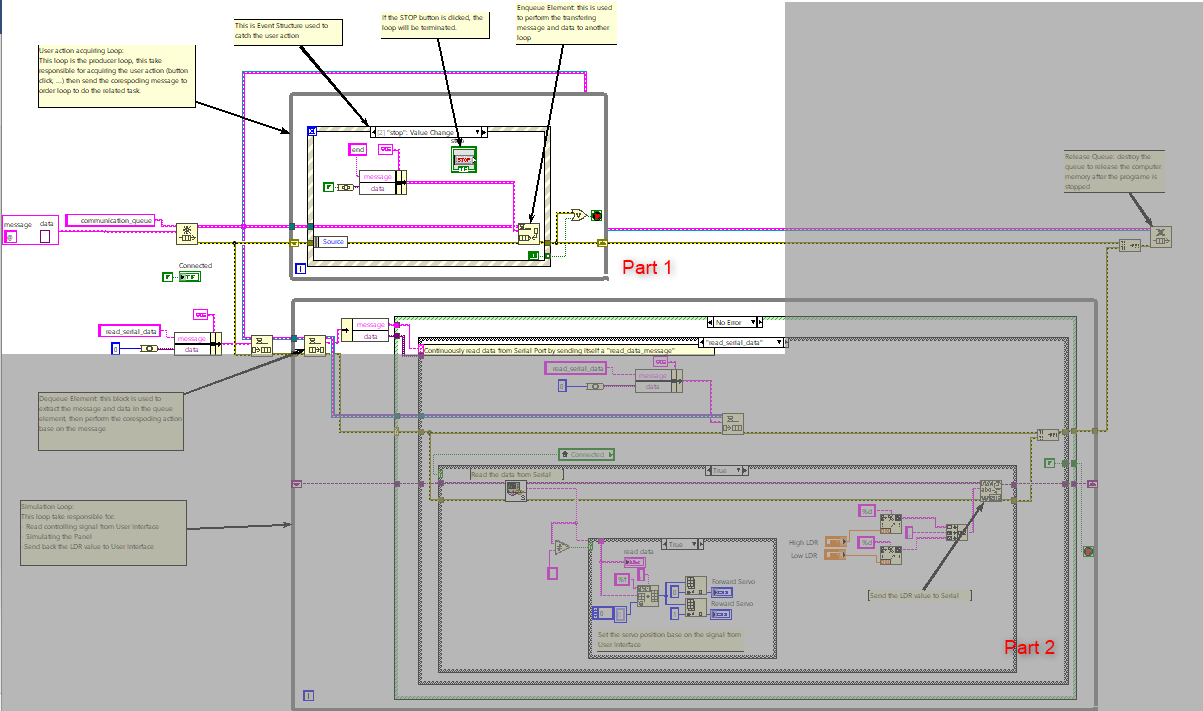


# Panel Simulation:



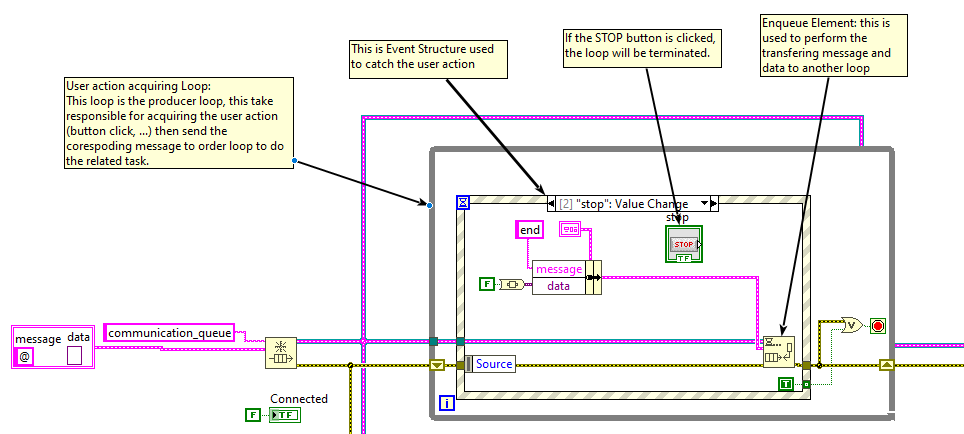
1. Serial Port Name:  
   The name of the Serial Port available for communication (to communicate to the hardware which control the motor or the software which simulate the motor).
2. Baud rate:  
   The baud rate of the Serial port, 9600 is mostly use due to its stable behavior.
3. Open button:  
   Click to open the port and start the communication.
4. Close button :  
   Click to close the port and stop the communication.
5. Connected:  
   The LED to indicate that the communication is ready to be used (send/receive data).
6. High LDR:  
   Simulated High LDR value.
7. Low LDR:  
   Simulated Low LDR value.
8. Forward Servo:  
   Indicate the forward servo position.
9. Reward Servo:  
   Indicate the reward servo position.
10. Read data:  
    Data read from Serial Port.
11. Forward Servo Position .
12. Reward Servo Position .
13. Stop button:  
    Click this button to stop the program.

**Diagram:**



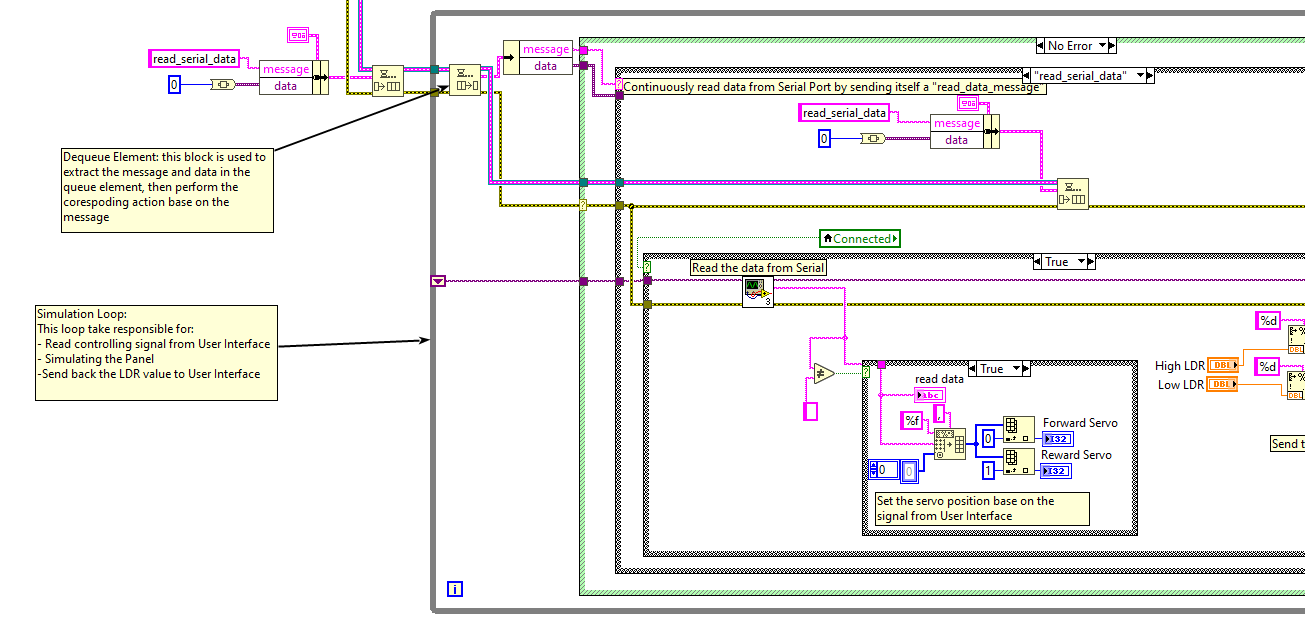
1. **Part 1:**

User action acquiring Loop:  
This loop is the producer loop, this take responsible for acquiring the user action (button click, ...) then send the corespoding message to order loop to do the related task.



1. **Part 2:**

Simulation Loop:  
This loop take responsible for:   
- Read controlling signal from User Interface  
- Simulating the Panel  
-Send back the LDR value to User Interface



**Overview** **Diagram:**

**Prj2\_Main.vi**

Port

Servo Control Loop

Desired Servo Position

**System\_2.vi (Panel simulation)**

Port

Simulation Loop

Panel Simulation

Serial port control

User interface acquiring loop

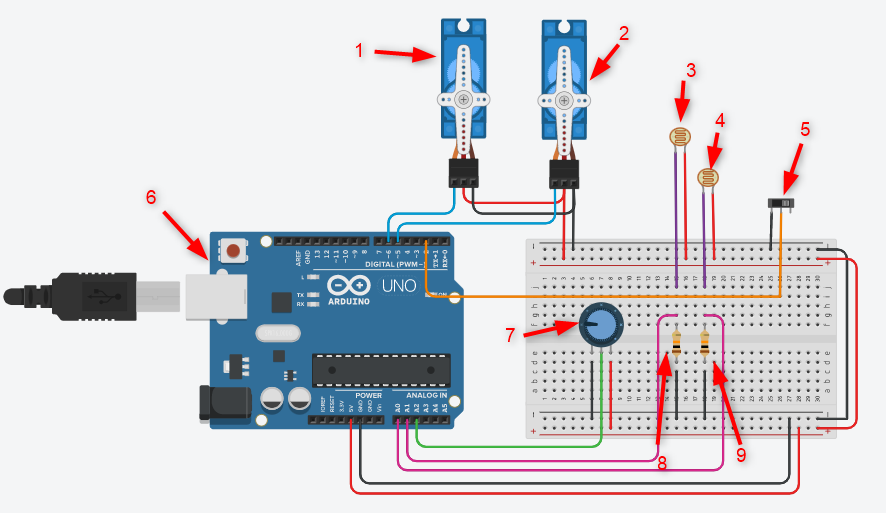
Communication Loop

User interface acquiring loop

Serial Bus

Real servo position

# Tinker CAD Diagram



1. Reward Servo
2. Forward Servo
3. Low LDR
4. High LDR
5. Mode switch to select Auto or Manual mode
6. Arduino Board
7. Potential Meter for control the Servo in Manual mode
8. Resistor support the Low LDR
9. Resistor support the High LDR

# Arduino code

1. **Assign alias to Arduino pin**

#define HIGH\_LDR\_PIN A0

#define LOW\_LDR\_PIN A1

#define POT A2

#define AUTO\_SWITCH 2

#define FORWARD\_SERVO\_PIN 5

#define REWARD\_SERVO\_PIN 6

1. **Assign alias to constants**

#define TRUE HIGH

#define FALSE LOW

#define SERVO\_SPEED 10

#define SERVO\_MIN\_POSITION 0

#define SERVO\_MAX\_POSITION 180

#define RUNNING\_CYCLE 1

#define AUTO\_MODE TRUE

#define MANUAL\_MODE FALSE

1. **Declare necessary variable**

Servo forward\_servo;

Servo reward\_servo;

long forward\_servo\_position = 90;

long reward\_servo\_position = 90;

1. **Setup function for setting up Arduino pin for read/write signal**

void setup(){

  forward\_servo.attach(FORWARD\_SERVO\_PIN);

  reward\_servo.attach(REWARD\_SERVO\_PIN);

  forward\_servo.write(forward\_servo\_position);

  reward\_servo.write(reward\_servo\_position);

  pinMode(AUTO\_SWITCH, INPUT\_PULLUP);

  Serial.begin(9600);

}

1. **Loop function for executing the Panel control and print out the system information (Control mode, servo position) to Serial Monitor.**

void loop() {

  // If user select Auto mode

  if (digitalRead(AUTO\_SWITCH) == AUTO\_MODE)

  {

    // read the analog value of High LDR and Low LDR

    int high\_ldr\_value = analogRead(HIGH\_LDR\_PIN);

    int low\_ldr\_value = analogRead(LOW\_LDR\_PIN);

    // if High LDR value greater than Low LDR, increase the forward servo degree and vin versa

    if (high\_ldr\_value > low\_ldr\_value)

    {

      if (forward\_servo\_position < SERVO\_MAX\_POSITION)

      {

        forward\_servo\_position = forward\_servo\_position + SERVO\_SPEED;

      }

    }

    else if (high\_ldr\_value < low\_ldr\_value)

    {

      if (forward\_servo\_position > SERVO\_MIN\_POSITION)

      {

        forward\_servo\_position = forward\_servo\_position - SERVO\_SPEED;

      }

    }

    reward\_servo\_position = 180 - forward\_servo\_position;

  }

  // If user select Manual mode

  else if (digitalRead(AUTO\_SWITCH) == MANUAL\_MODE)

  {

    // read the potential meter value then calculate the servo

    float pot\_value = float(analogRead(POT));

    forward\_servo\_position = pot\_value\*180/1023;

    reward\_servo\_position = 180 - forward\_servo\_position;

  }

  // from the calculated position, output that position to the servo

  forward\_servo.write(forward\_servo\_position);

  reward\_servo.write(reward\_servo\_position);

  // Print out the system information (control mode, servo position)

  if (digitalRead(AUTO\_SWITCH) == AUTO\_MODE)

  {

    Serial.print("AUTO");

  }

  else

  {

    Serial.print("MANUAL");

  }

  Serial.print('\t');

  Serial.print("forward\_servo: " + String(forward\_servo\_position));

  Serial.print('\t');

  Serial.print("reward\_servo: " + String(reward\_servo\_position));

  Serial.print("\t\r\n");

  // delay a little bit for safety functionality

  delay(RUNNING\_CYCLE);

}