

# Surveillance bot

## Problem Statement

To control a differential drive through PS2 controller and camera interfacing on a Raspberry Pi.

## Explanation

First, we make a chassis which having 4 wheelers differential drive mechanism by using PWM and PID algorithms and which can be controlled by PS2 controller where communication between microcontroller and PS2 controller takes place using SPI communication protocol.

Second, we will interface web camera using Raspberry Pi where Raspberry Pi is a microprocessor. It having its own OS (Operating System) named Noobs and Raspbian.

## Components

* ATmega128
* Raspberry Pi
* USB Web Camera
* Cytron (Motor Driver)
* 4 DC Motors (12 volts)
* IC7805 (Voltage Regulator)
* Or some other tools

## PS2 Controller

PS2 controller is the gaming controller which having digital (Pressure buttons) and analog (Joy Sticks) and it works on SPI communication. After joining appropriate Pins of PS2 with ATmega128, we need to configure it which we do through programming. For we are using ATMEL software.

## Raspberry Pi

Raspberry Pi is a one microprocessor containing RAM, connecting modules like Bluetooth, Wi-Fi and LAN and some ports and pins for connecting external devices. For our project we are using **Raspberry Pi 3b+** model which is the latest one.

For programming PI, we require a desktop having HDMI port .We avoided using HDMI port and accessed PI through Ethernet cable and wifi.

For live streaming of surrounding environment, we are using USB web camera which connect with Raspberry Pi and then using some library and packages we will configure Raspberry Pi as live streaming server. After all configuration we can see live video on web-browser using IP address of PI. (But for this PI and device must be connected through same Wi-Fi network.)

## Cytron

It is a one type of motor drive which can handle large variation in input current.

And heat generation in this is less than others Motor driver.

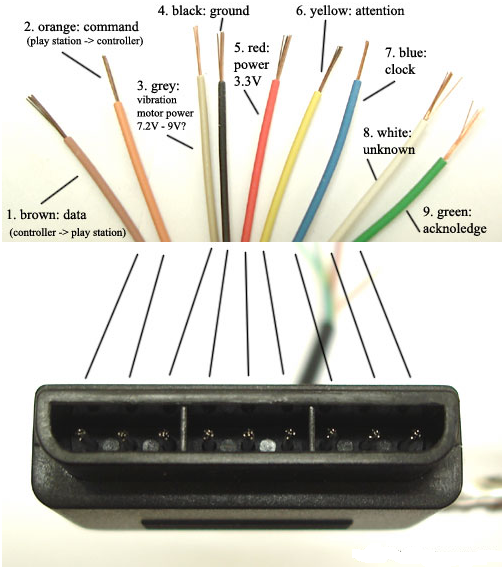
## Differential Drive

As name suggest, Bot will change direction and speed due to speed difference between left side and right side’s motors speed (RPM). This can be archive by uses of PWM (Pulse Width Modulation). For getting stable speed (RPM of Motors), we are PID (Propositional, Integration & Diffraction) algorithm.

**PS2 CONTROLLER**

****

The PS2 console gamepad has 9 wires- 5 communication lines, VCC,GND, vibration motor power, and a reserved line for future use.

[](http://4.bp.blogspot.com/-xipuAaFkcvk/VADDDcHE2jI/AAAAAAAAAIU/WKvVWTcnOj4/s1600/1.png)

**Pin Color Name Description**

**1 Brown**

**DATA :**This is the signal from controller to host. It is an 8-bit serial transmission synchronous to the falling edge of the clock. **THIS IS CONNECTED TO MISO of Atmega128. This is an [open collector output](http://en.wikipedia.org/wiki/Open_collector) and requires a pull-up resistor (1 to 10k, maybe more). (A pull-up resistor is needed because the controller can only connect this line to ground; it can't actually put voltage on the line).**

**2 Orange**

**Command :**This is the signal from the host to the controller. Again, it is an 8-bit serial transmission on the falling edge of the clock `.**THIS IS CONNECTED TO MOSI of Atmega128**

**3 Grey**

**7.6V :**Vibration Motors Power: 6-9V.

**4 Black**: **Ground**

**5 Red**

**VCC :**Variable power from 5 V down to 3V.

**6 Yellow**

**Attention :**This signal is used to get the attention of the controller. This signal will go low for the duration of a transmission .**THIS IS CONNECTED TO SS' of Atmega128**

**7 Blue**

**Clock** : 500kHz **THIS IS CONNECTED TO SCK (PIN PB7) of Atmega128**

**8 White**

**Not Connected:** Reserved for future use.

**9 Green**

 This signal is low for at least one clock period after each 8 bits are sent and ATT is still low.

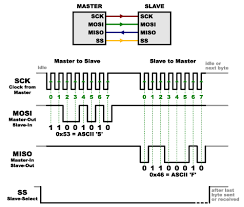
Connect the Clock line and data line as directed above **should be pulled up to VCC with a 10K resistor.**

**SPI PROTOCOL – SERIAL PERIPHERAL INTERFACE**

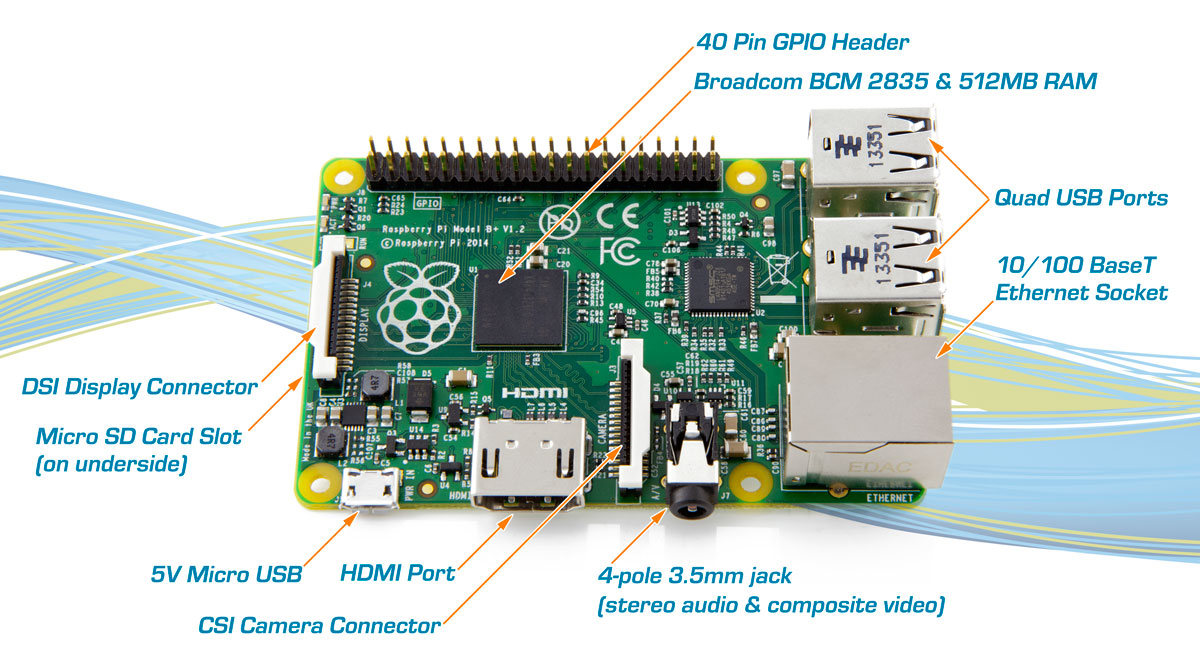
SPI(Serial peripheral interface) protocol is used to communicate between microcontroller and any other sensor or controller.SPI is used to talk to a variety of peripherals, such as

* Sensors: temperature, pressure, [ADC](https://en.wikipedia.org/wiki/Analog-to-digital_converter" \o "Analog-to-digital converter), touchscreens, video game controllers
* Control devices: [audio codecs](https://en.wikipedia.org/wiki/Audio_codec" \o "Audio codec), digital potentiometers, [DAC](https://en.wikipedia.org/wiki/Digital-to-analog_converter" \o "Digital-to-analog converter)
* Camera lenses: [Canon EF lens mount](https://en.wikipedia.org/wiki/Canon_EF_lens_mount" \o "Canon EF lens mount)

For spi there are mainly four pins on the microcontroller:MOSI(master out slave in),MISO(master in slave out),SCK(clock),SS(slave select). In spi there will be one master and there can be multiple slaves depending upon the system, normally the SS pin is kept HIGH and to select that slave to communicate we LOW the SS pin associate with it.master initialize the communication between them and communication between them is done at the clock set by master. To see the different registers in spi and their functionality and how to use it see the data sheet of Atmega 128.



**RASPBERRY PI SETUP**



1. Download Raspbian Jessie OS from here: <https://www.raspberrypi.org/downloads/raspbian/>
2. Take a new sd card or format an old sd card using SD card formatter.
3. Flash image of downloaded Raspbian OS in the sd card using Etcher.
4. Create a blank text document “ssh” without any extension or Using command prompt type :echo>E:/ssh
5. Remove the SD card and insert it into the raspberry pi SD card slot.
6. Connect your pi to your laptop using ethernet cable (there is an ethernet port on pi).
7. Supply power (5V) to the raspberry pi using the microUSB cable(connected to laptop or powerbank)
8. Open command prompt and write the following command to get the IP’s of all connected networks and note down the Dynamic IP address returned: “arp -a”
9. To check which is the required IP address, type command : ”ipconfig” in command prompt.(The IP corresponding to ethernet to ethernet is our required IP)
10. To check whether transmission of data can occur through this ip,write command: “ping IP Address”(eg. 192.168.43.206 or you can also write “ping raspberrypi.mshome.net”)
11. Download PuTTy using link:
12. Open PuTTy ;Type in your IP address for Pi ;select connection type as SSH ;Then go to category/SSH/X11, enable X11 forwarding and set X display location as ‘localhost0’
13. This will open command prompt of raspi, which will ask for username and password.By default it
14. For Remote desktop accessing raspberrypi :In this cmd prompt download xrdp file using command :”sudo apt-get install xrdp”.
15. Open Raspbian OS using Remote desktop(on windows).
16. Open terminal.
17. Update and upgrade your raspberrypi using command: sudo apt-get update && sudo apt-get upgrade.

**LIVE STREAMING USING RASPBERRY PI**

1. Installing Motion package:

Open terminal.  
 Enter commands: sudo apt-get install motion  
 sudo nano /etc/motion/motion.conf

-turn on daemon

-turn off localhost in livestream settings

sudo nano /etc/default/motion

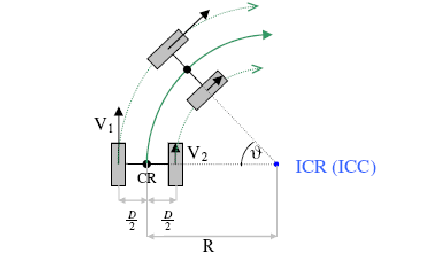
-turn daemon on

1. Connect USB camera to USB port of raspberrypi
2. Enabling USB webcam:sudo raspi-config  
    - In Interfacing options , enable camera.
3. Start Live streaming:  
    -sudo service motion restart
4. Open web browser using the same network as raspberrypi to view live streaming-search :” IP address:8081”
5. To stop Live Streaming:  
    -sudo service motion stop
6. To shutdown raspberrypi:  
    -sudo poweroff

**Differential Drive:**

Differential drive consists of two drive wheels mounted on A common axis and control of each wheel is independent either forward or backward. By varying the velocities of two wheels, we can vary the trajectories the robot takes.

The motion of the wheels causes the motion vector which is the sum of the independent velocity vector.

.

To implement differential drive we are using PWM to regulate the speed of motors and PD to reduce the errors. For example if we want to make turn of *w* angle on the right hand side we will decrease the speed of right hand side motor to say by 50% and left hand side motor to say by 30% which will cause the right side motor to have less rpm comparitively to left side motor which causes the bot to move on right side.

To determine the speed of motors to take certain angles we are going with some expected data-sets.We have used cytron as motor driver.

• UART COMMUNICATION

• Introduction :-

• UART stands for Universal Asynchronous Receiver Transmitter. It is basically just a piece of computer hardware that converts parallel data into serial data.

• • Serial Communication Protocol :- serial communication is the process of

sending/receiving data in one bit at a time.

• Two wire communication protocol

• Data Format and Transmission speed are configurable.

Some of the main features of the AVR USART are:

• Full Duplex Operation (Independent Serial Receive and Transmit Registers)

• Asynchronous or Synchronous Operation

• Master or Slave Clocked Synchronous Operation

• High Resolution Baud Rate Generator

• Supports Serial Frames with 5, 6, 7, 8, or 9 Data bits and 1 or 2 Stop Bits

• **Asynchronous :-**

It has no clock signal require For the Receiver to synchronize the validate the data from transmitter to receiver.

Modes of Operation

The USART of the AVR can be operated in three modes, namely-

1. Asynchronous Normal Mode 2. Asynchronous Double Speed Mode 3. Synchronous Mode Asynchronous Normal Mode

In this mode of communication, the data is transmitted/received asynchronously, i.e. we do not need (and use) the clock pulses, as well as the XCK pin. The data is transferred at the BAUD rate we set in the UBBR register. This is similar to the UART operation. Asynchronous Double Speed Mode

This is higher speed mode for asynchronous communication. In this mode also we set the baud rates and other initializations similar to Normal Mode. The difference is that data is transferred at double the baud we set in the UBBR Register.

Setting the U2X bit in UCSRA register can double the transfer rate. Setting this bit has effect only for the asynchronous operation. Set this bit to zero when using synchronous operation. Setting this bit will reduce the divisor of the baud rate divider from 16 to 8, effectively doubling the transfer rate for asynchronous communication. Note however that the Receiver will in this case only use half the number of samples (reduced from 16 to 8) for data sampling and clock recovery, and therefore a more accurate baud rate setting and system clock are required when this mode is used. For the Transmitter, there are no downsides.

• UART CONNECTION

In UART, Data flows from TX pin of the transmitter to RX pin of the Receiver. So that Tx pin of DEVICE 1 is directly connected to Rx pin of the DEVICE 2. and As to Achieve common Reference both the Transmitter and Receiver are connected through common Ground.

UART achieve data synchronization between Transmitter and Receiver by two simple mechanism

: • **Data Synchronization in UART :-**

1. Baud rate. II. Start and Stop bit.

**I) BAUDE RATE :-** First both the devices are must agree on timing parameters before hand which can be achieved by configuring baud rate.

• BAUD RATE is rate of data transmission in serial communication and it is expressed in bits per second(bps).

• There are few standard baud rate are defined which can be configured in both the devices . I.e. 2400, 4800, 9600 etc.

• **II)Start and Stop bit. :-** UART uses two special bits. One at the beginning and Other at the end of each data byte. In UART Terminology, this special bits are called as start and stop bits.

• UART DATA FRAMING :-

The ideal no data state is High voltage or powered on in which line is held high to show that the line and transmitter are not damaged.

Each frame consist of a start bit, Data bit , parity biy(possibly) and one or more stop bits.

Start bit signals the receiver that new data is coming.

Next 5 to 9 bits, depending upon the configuration represent the data.

If the parity bit is configured, it will be placed after all the data bits.

Then next one or two bits are always marked or logic high state, which is called stop bits, that signals the receiver, all the character is completed. All character will be transmitted one by one in the same format.

And the parity bit is used by receiver for error detection, If thre is data has changed during the transmission.

• UART Configuration :-

One can configure UART in three different ways, Depending one the application.

**1) SIMPLEX :-** Simplex is one directional configuartion with no provision for the receiving device to send the information back to transmission.

**2)HALF DUPLEX:-** Half Duplex can happen in both direction but not at a time.

**3) FULL DUPLEX :-** Full duplex can be transmit and receive data at the same time.

There are some merits as well as demerits of UART Protocol. Let’s see.

Advantages:-

• Only uses two wires.

• No clock signal is necessary

• Provide error detection by parity bit check.

• Cost and size will be lesser to the parallel communication.

• Disadvantages

• The size of data frame is limited to a maximum of 9 bits.

• Doesn’t support multiple slave or multiple master system.

• Limited speed is the bottleneck for the application which require higher data of transmission.