

# Position control of Servo

## Gesture based control of servo motor

Harish Guda

Shehzaman S. Khatib

Siddharth S.



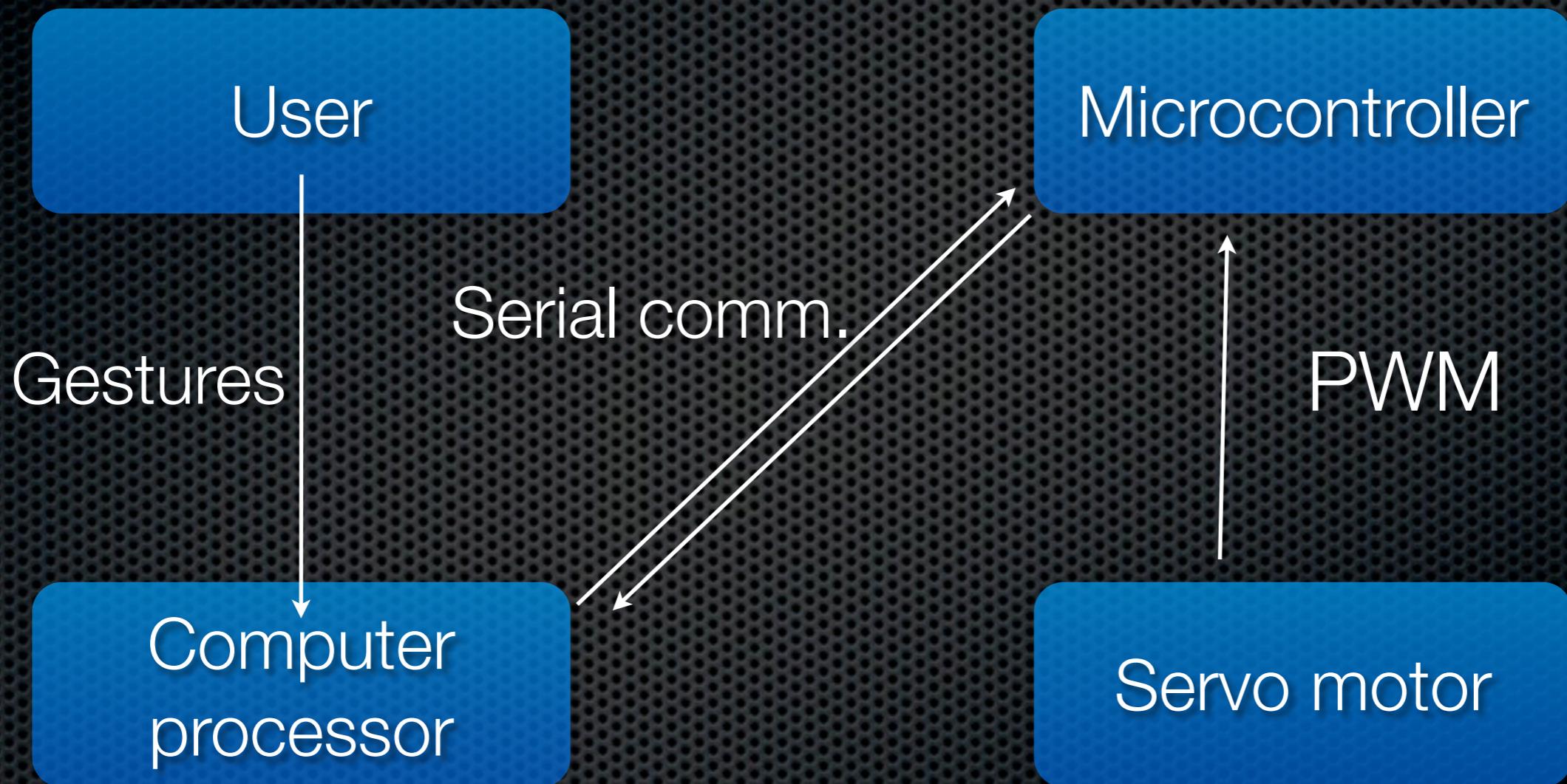
# Objective

To control the position of a servo motor using hand based gestures.

# Modules

- Image processing.
- Serial communication between computer's processor and the micro controller.
- Servo positioning and control using pulse width modulation (PWM).

# Modules



# Image Processing

- Realtime signal processing of multiple images.
- Snapshots are taken approximately 24 times a second.
- The snapshot is processed to obtain the gesture that is being shown at that instant.
- Three gestures are being recognized in the current version, which are mapped to three positions of the servo.

# Image Processing

- Library used - OpenCV ([opencv.willowgarage.com](http://opencv.willowgarage.com))
- Standard C++, compiled with the GNU compiler collection (gcc or g++, v4.2.1)

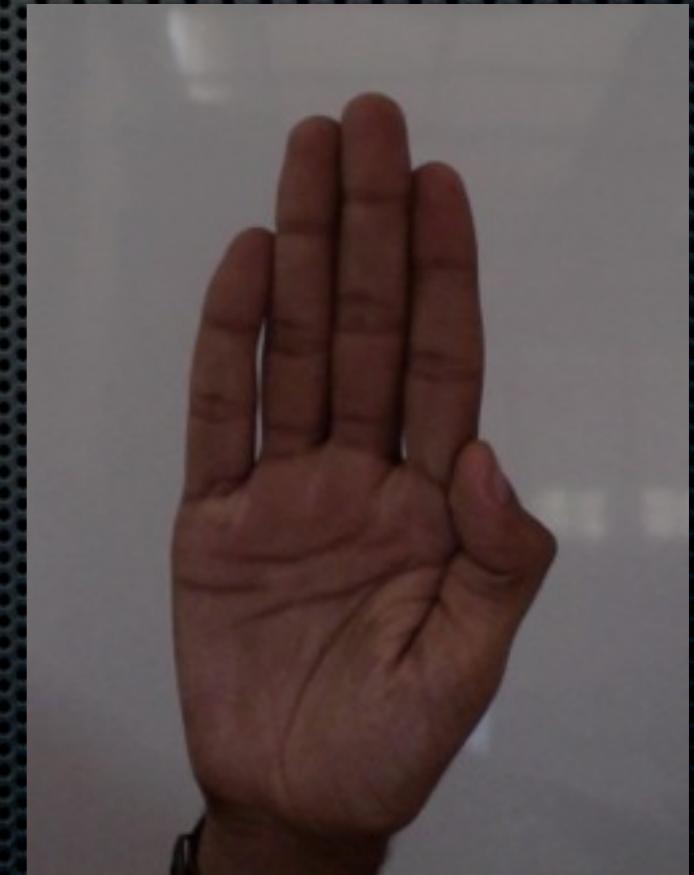
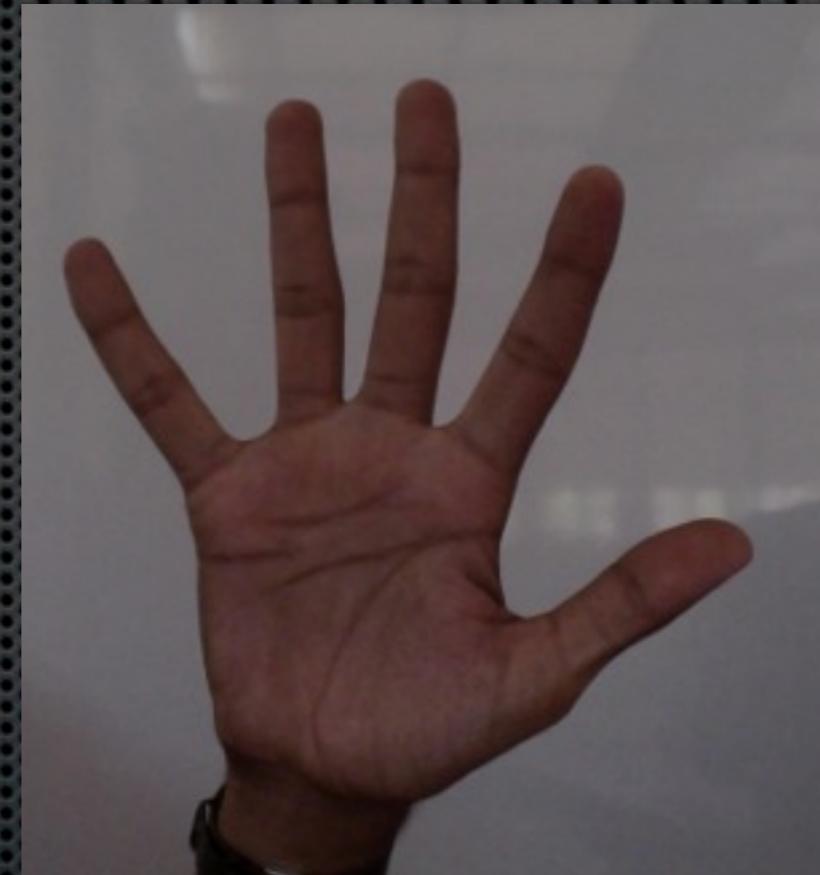


# Algorithm - an overview

- Background subtraction and hand recognition.
- Tracking and prediction.
- Image processing and recognition of gesture.
- Eigenfaces method to recognize the gesture.
- Mapping to corresponding action.

# Gestures

- Three gestures are recognized in the current version.



# Tracking

- Two important parts to this : Camshift algorithm and Kalman filter.
- Kalman filter predicts the position of the hand in the next frame, given the present velocity and position.
- Camshift algorithm does the actual measurement of the bounding box.
- This value is passed back to Kalman class to make corrections.

# Gesture recognition

- Method called eigenface - first method used for face recognition, in 1987 by Sirovich and Kirby.
- Eigenfaces are sets of eigenvectors.
- Eigenvectors :  $A\mathbf{v} = \lambda\mathbf{v}$ .

# Gesture recognition

- The program is trained to recognize a gesture by a sample set of 10 images.
- The average value of this images is compared with the frame in question.
- Recognition: Project the image onto a different vector face generated by performing a mathematical process called principal component analysis on the sample images.

# Gesture recognition

- Height of the bounding box > 2.3 times width.
- Height of the bounding box ~ 1.3 to 2.3 times width.
- Height of the bounding box < 1.3 times width.

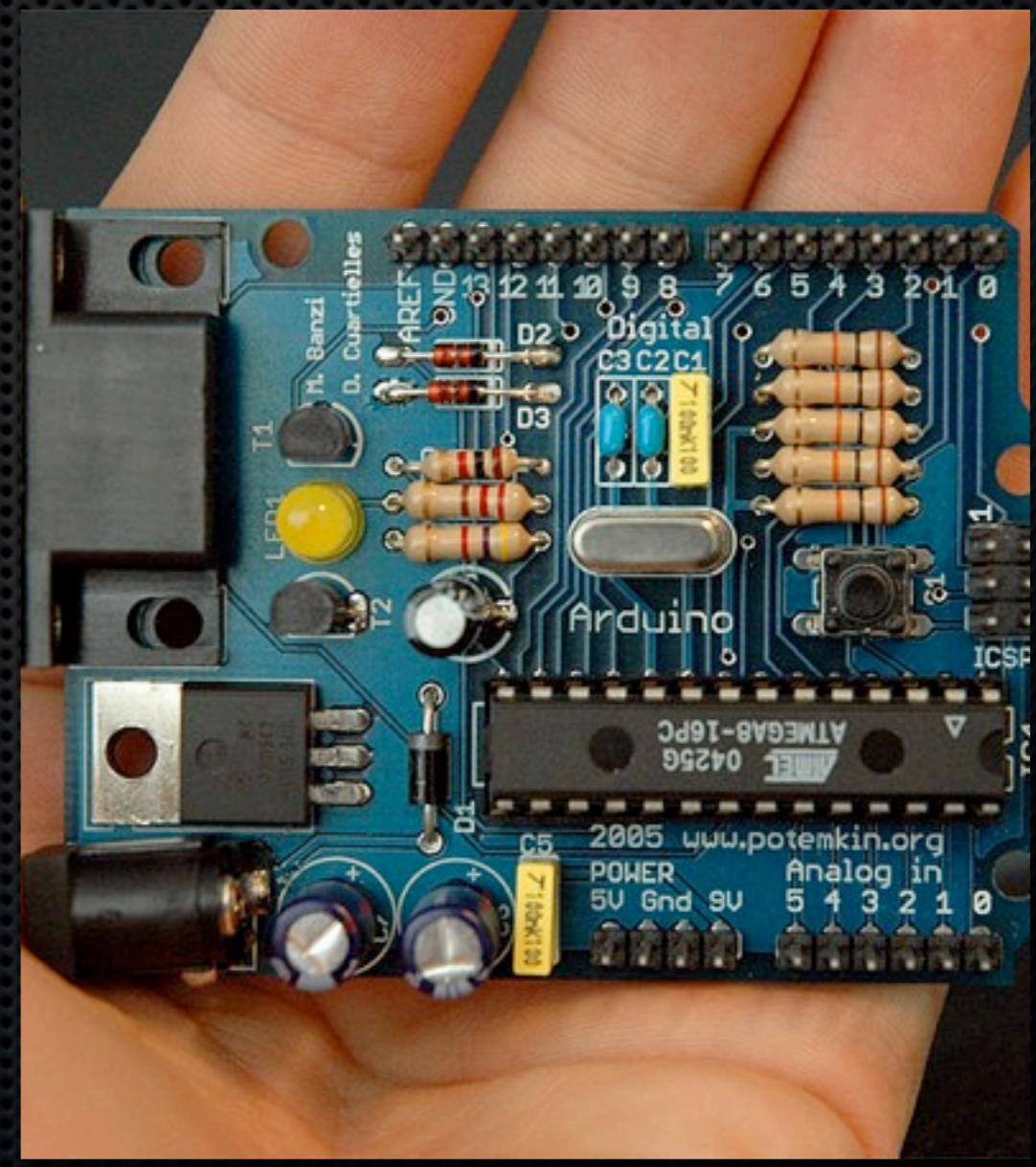
Numbers are empirically calibrated.

# Mapping

- The different gestures are finally mapped to a set of actions.
- This is where the interface between the computer's processor and the microcontroller is invoked.

# Arduino

- Arduino is an open-source single board microcontroller.
- Atmel AVR processor onboard (Atmega 128).
- Programming interface based on C++.



# Serial port communication

- Process of sending data, one bit at a time, sequentially over a communication channel or bus.
- Parameters are :

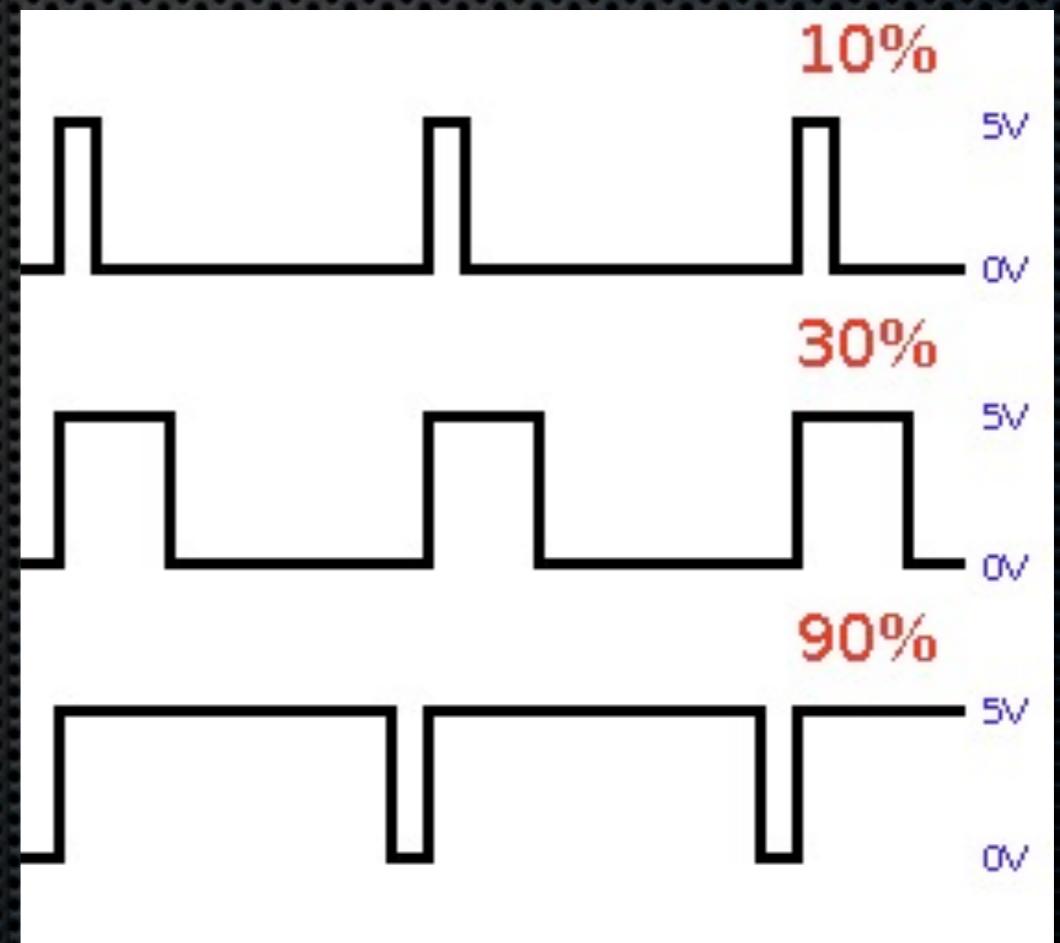
Baud rate	Number of bits	Stop bits
Parity	Hardware flow control	COM port

# Data sent

- The microcontroller is programmed in such a way that it keeps listening on the serial port for incoming data.
- The useful data that has to be sent to the controller are pre-defined characters, that define the state of the servo.
- These characters are mapped to servo positions, and the corresponding signal is generated.

# Servo control signal

- Pulse-width modulation (PWM), or pulse-duration modulation (PDM) is a commonly used technique for controlling power to inertial electric devices.
- Different duty cycles are obtained by changing the on and off time of the signal.



# Servo control signal

- The servo has an inbuilt controller that translates a particular PWM value into a specific motor angle.
- For our purpose, angles of  $0^\circ$ ,  $90^\circ$ , and  $180^\circ$  are used, and are mapped to the three gestures.

# Error analysis

- Servo positioning is very accurate ( $\sim \pm 0.1^\circ$ ).
- It has its own inbuilt circuitry, that specifies and necessitates signals of only a particular frequency, and amplitude.
- Corrections for deviations and deflections due to load are taken care of by the inbuilt controller, with the aid of position feedback potentiometer.

# Applications

- The primary aim of the project is to demonstrate the scope of gesture based control.
- The servo motor is just an example of an end actuator.
- The servo can be replaced by a wide range of actuators, leading to several applications.

# Applications

Gear changing in  
Automatically guided  
vehicles

Assists physically  
disabled by making  
communication easier

Saves human-computer  
interaction time

Simplifies use of HMI

Wide range of gestures  
can be integrated

Complements speech  
recognition to automate  
processes

# References

- OpenCV documentation
- <http://opencv.willowgarage.com/wiki/>
- <http://www.futaba-rc.com/servos/analog.html>
- <http://arduino.cc/en/Reference/HomePage>
- <http://arduino.cc/en/Main/ArduinoBoardUno>
- Thanks to K. Vignesh, for helping us with OpenCV.

# Acknowledgements

Sincere thanks to

- Prof. Singaperumal
- Prof. Manivannan
- Prof. Somashekhar Hiremath
- Prof. Ashish Kumar Sen

Thank you.

Its demo time!