Python API for Mobile Robot Control

Progress Presentation-1

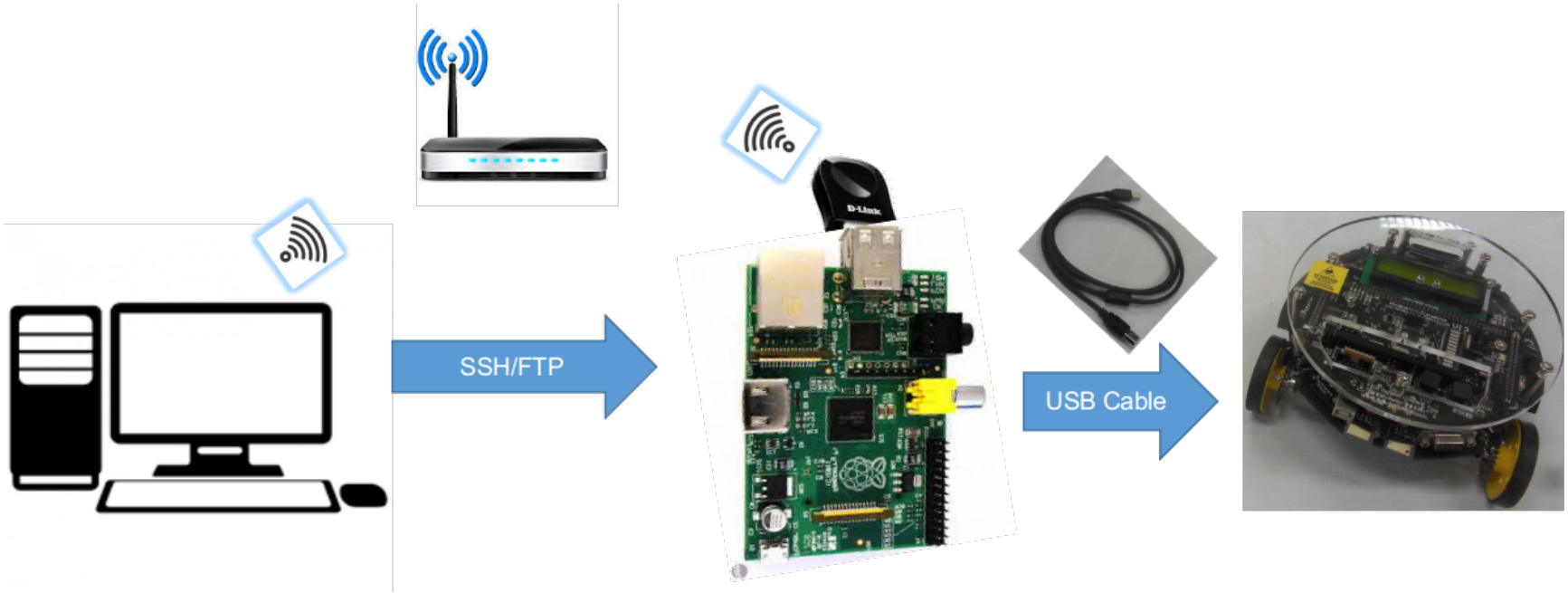
Parin Chheda (153076005)
Saurav Shandilya (153076004)
Group-10

Prof. Prabhu Ramachandran
Prof. Madhu Belur
Prof. Kumar Appaiah

Objective

1. A Python API to control the different peripherals of the μcontroller
2. Provide the user with a option of register level access of the μcontroller
3. Allow the user to design an application without learning a new language and thoroughly knowing the architecture of the controller.

System Architecture



Milestone Achieved

1. Robot-end firmware - written in embedded C
2. Serial Communication between Raspberry Pi and Robot
3. Function for configuring all IO Ports and Pins of microcontroller
4. Implemented and tested code for Buzzer and BarLED.
5. Test code for Port and Pin configuration function

User code snippet

```
from set_register import *
import time
import serial_connection as sc
sc.serial_open()

###Configure ports###
#set port pin direction
config_register(DDRJ,PIN1|PIN2|PIN3,set_pins)
#intial value logic 0
config_register(PORTJ,PIN1|PIN2|PIN3,reset_pins)

for i in range(0,5):
        #turn on Leds
        config_register(PORTJ,PIN1|PIN2|PIN3,set_pins)
        time.sleep(0.5)
        #turn off leds
        config_register(PORTJ,PIN1|PIN2|PIN3,reset_pins)
        time.sleep(0.5)

sc.serial_close()
```

Implementation

Data Packet Format

set/rest	Function id	Register id	Pin value
1-bit	7-bit	8-bit	8-bit

set/reset (1-bit): This bit decides whether value a register bit is set high or reset to low

Function id (7-bit): These bits select various μcontroller peripherals Example: IO,ADC,Timers,I2C etc

Register id (8-bit): These bits select registers present to access μcontroller peripherals. Example: DDRA,PORTA, ADCSRA,ADMUX etc

Pin Value (8-bit): These bits select value to be written on 8-bit registers

Future Work

- Develop and test object-oriented implementation
- Access following peripherals for μcontroller
  - Timers
  - ADC
  - Interrupt
  - I2C
- Improve data packet by incorporating checksum, end of packet payload to existing system.
- Design PyQT GUI for making SFTP connection to Raspberry Pi from PC
- Provide higher level abstraction for peripheral devices