

Architecture for Single-Process Systems

Lotfi ben Othmane

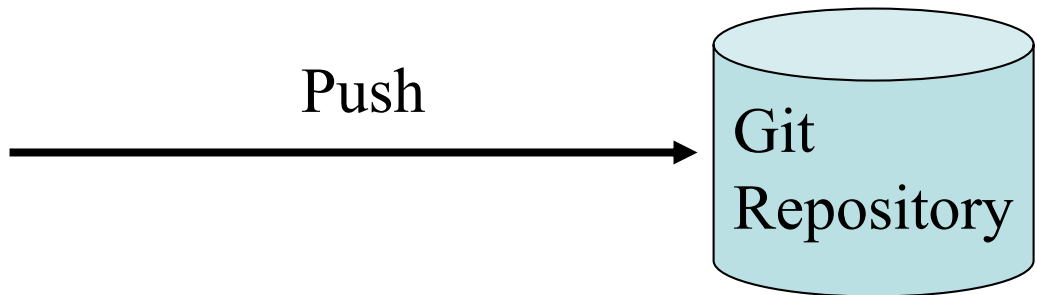
Question

Do the users need to share resources?
e.g., data or physical objects

Motivating Example

Assume that you installed Git client and created a repository

```
1
2
3 """
4 Created on December 19, 2018
5
6 Author: Lutfi ben Othmane
7 """
8 import csv # to read the csv file
9 import sys
10
11 #import random as rnd # for plot styling
12 #import numpy as np # data science package
13 from numpy import vstack,array
14 from numpy.random import rand
15 from scipy.cluster.vq import vq, kmeans, whiten
16 import copy
17 import collections
18 import matplotlib.pyplot as plt
19 import numpy as np
20 from scipy.stats import stats
21 import gc
22
23 count_learnmethod=[0]*7
24
25 """
26 This method returns the list of selected preferred learning methods for a given student
27 """
28 def getLearnMethod(Listmethods):
29     methods=[0 for i in range(7)]
30     ln = listmethods.split(',')
31
32     for i in range(len(ln)):
33         methods[int(ln[i])-1]=1
34         count_learnmethod[int(ln[i])-1]=1
35
36     # return an array that marks the used methods
37     return methods
38
39 """
40 This method returns
41 """
42 def EvaluateCognitiveLevel(level, CognitiveValue):
43     isCorrect = 0 # This indicates whether the question was correct or not
44
45 """
```



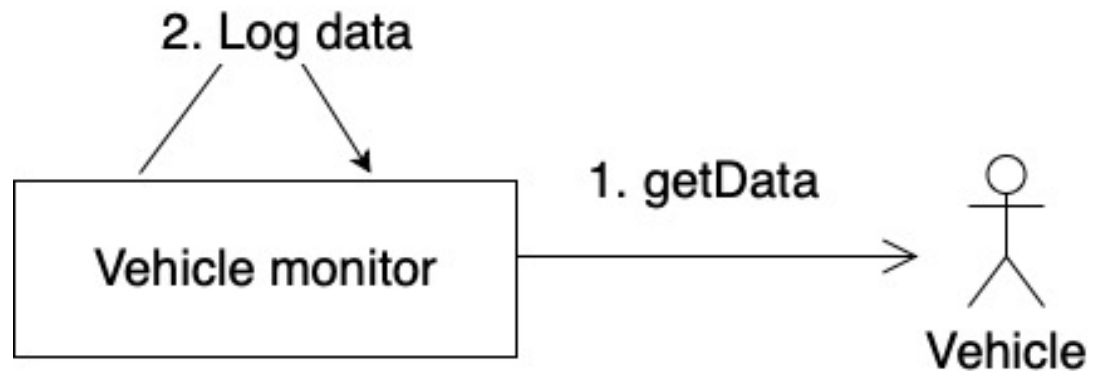
Does the Git client need to share information with another component of the system?

Motivating Example

Want to receive the data from the car and store them in a file.

The data of interest are

- Speed
- Fuel
- Engine errors
- Frequency of breaking



Single-Process Applications

- We can deploy a single-process app as one process.
 - It could be designed as one component

Architecture references for single-process software

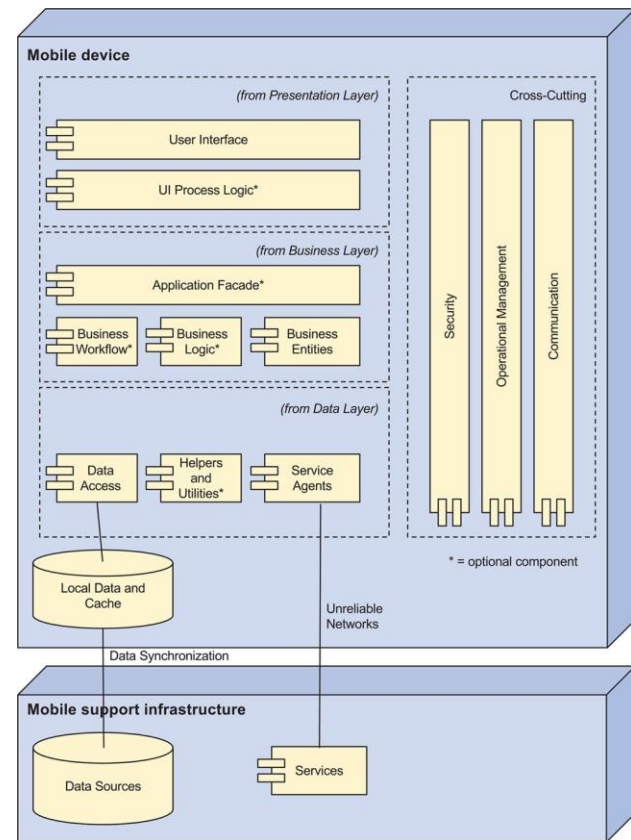
1. Desktop application
2. Embedded system
3. Mobile application
4. Service/ cron

Desktop Applications

- Operate on one device, e.g., desktop
- Examples: PowerPoint, word, etc.
- They are independent
 - They may use local database for e.g., storage
- The software needs to be installed on the client

Mobile Applications

- Mobile applications are typically executed on handheld devices.
- May use remote services
- May use a local database



Mobile Applications

- Cases for using it:
 - The application should/must run on handheld device
 - It is ok to have limited local resources
 - It is ok to have unreliable communication

Architecture References for Single Process Software

Reference architecture

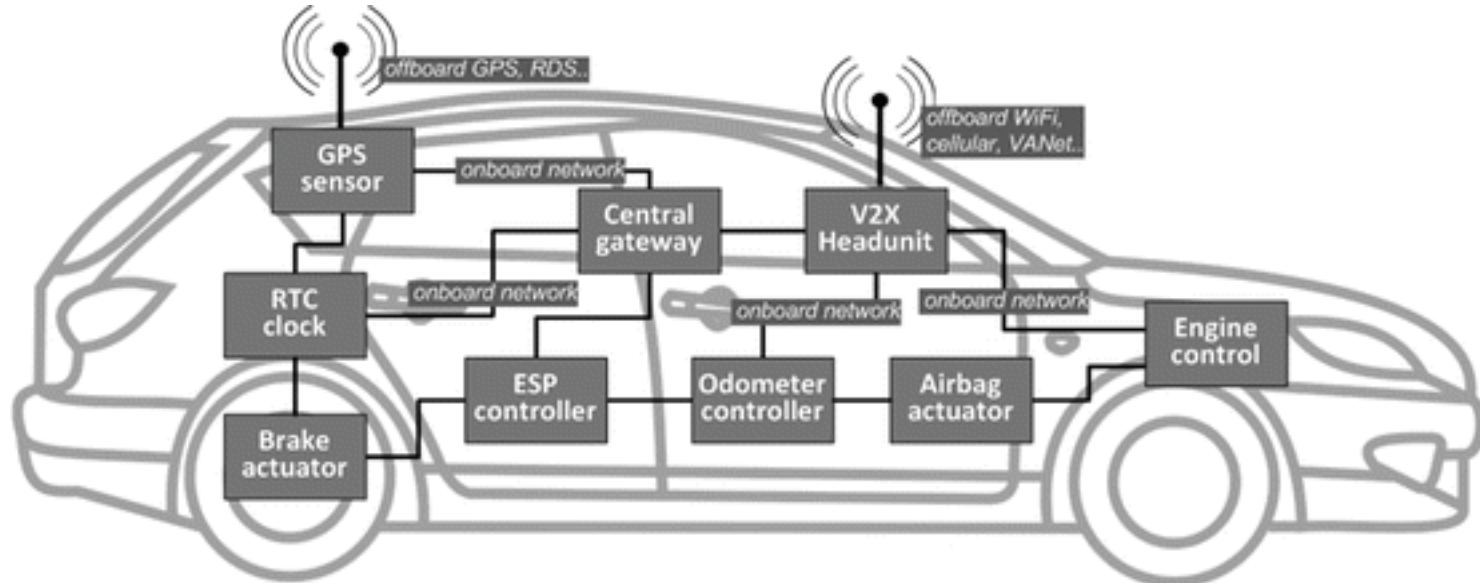
provides an overall logical structure for a particular type of applications

Possible architecture references

1. Desktop application
2. Embedded system
3. Mobile application
4. Service/ cron

Embedded System

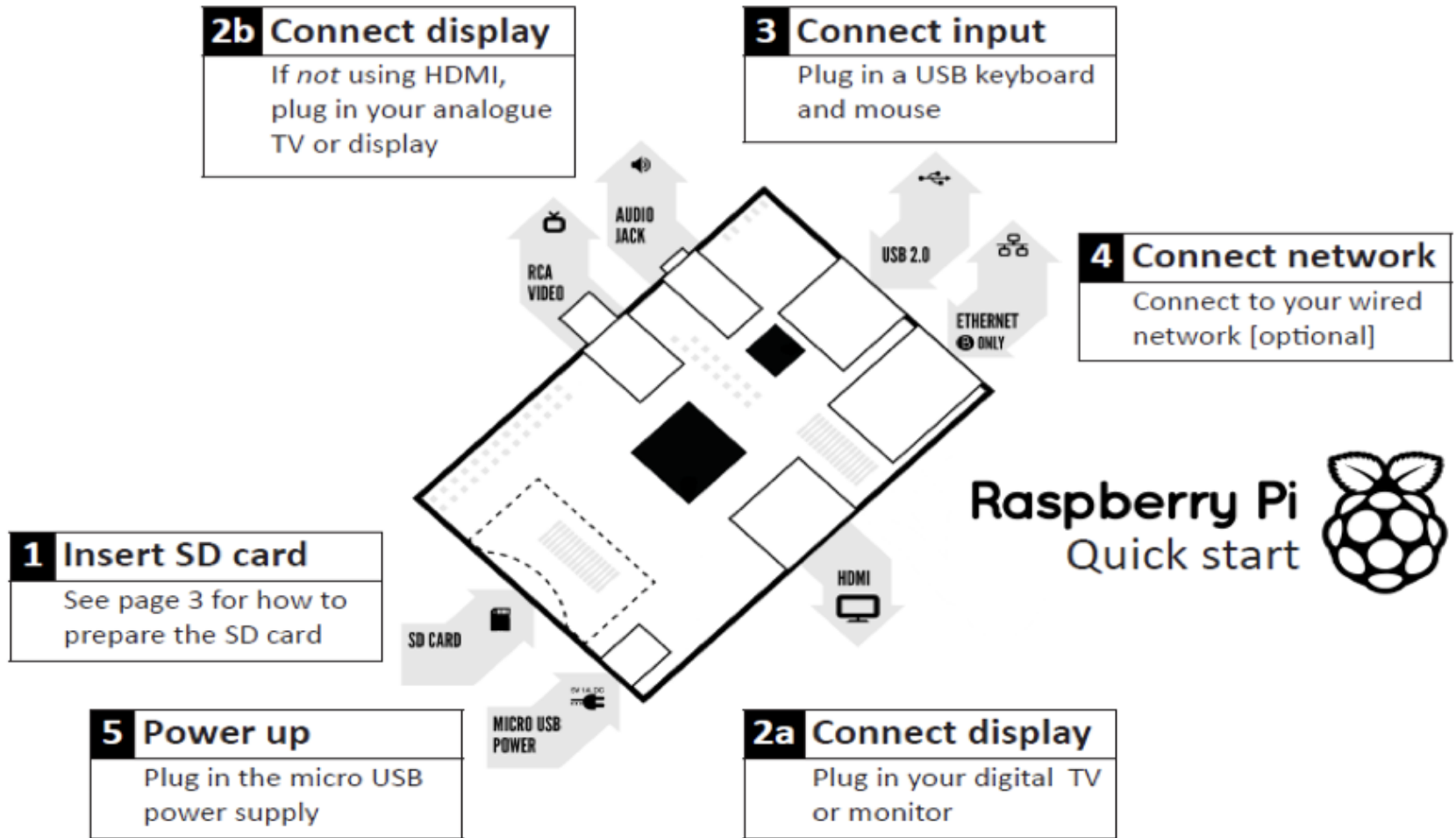
Embedded software is computer software, written to control machines or devices that are not typically thought of as computers, commonly known as embedded systems.



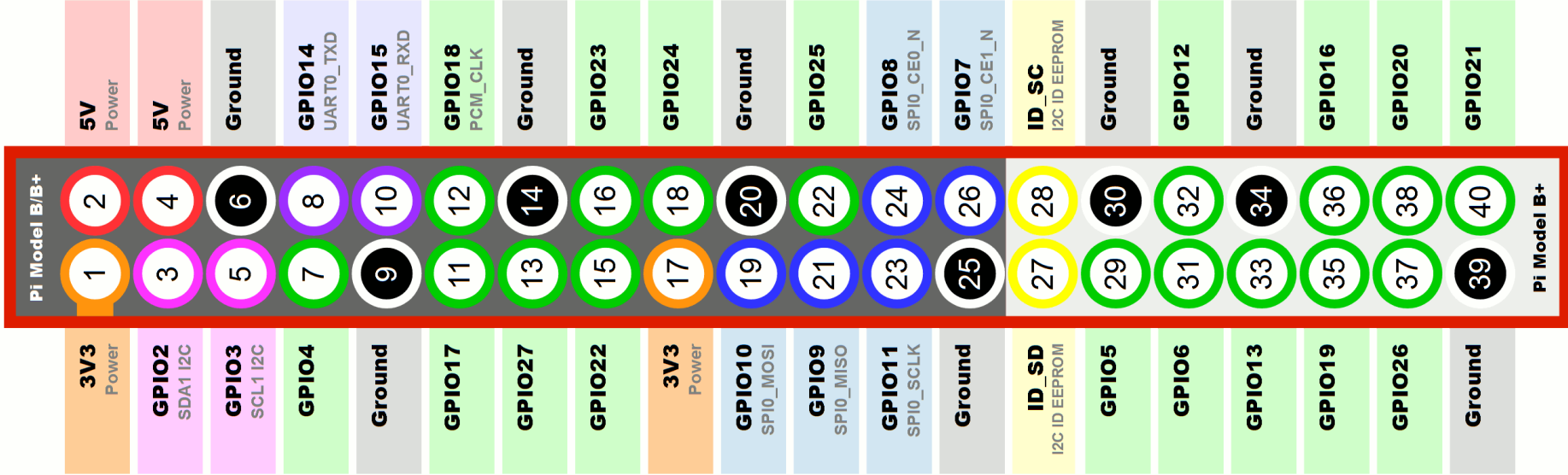
Example: Raspberry Pi 3 Model B+



Basic Hardware Connections



Basic Hardware Connections



Setting Up the Pi

- Put your SD card into your computer or SD card reader.
- Download NOOBs.
- Pick the “offline and network install” option.
- You may need to format your SD card as FAT.
- Extract the ZIP file and copy the entire contents of the folder to your SD card. Then, eject your SD card and insert it into the Raspberry Pi.
- See video for the full setup:
<https://www.raspberrypi.org/help/noobs-setup/2/>

Remote login

- Have ssh enabled for remote access
- `pi@raspberrypi ~ $ sudo raspi-config`
- Go to “interfacing options” and enable SSH.
- Use command “ifconfig” to get your IP address
- Remote command “ssh pi@[my IP]”
- You can use VNC too

First Program – Python code

Create a file: nano helloworld.py

```
## Content
```

```
Print("Hello Word")
```

```
##end
```

Execute: python helloworld.py

Python Basics

```
words = ['cat', 'window',  
'defenestrate']
```

```
for w in words:
```

```
... print(w, len(w))
```

```
> cat 3
```

```
> window 6
```

```
> defenestrate 12
```

List initialization

For statement

Python Basics

<https://docs.python.org/3/tutorial/controlflow.html#if-statements>

You will not write much python code for this class

Second Demo – Control of Lamp

```
import RPi.GPIO as GPIO  
import time
```

← Used libraries

```
relay_pin = 26
```

```
#relay_pin = 20
```

← Pins to use

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setwarnings(False)
```

```
GPIO.setup(relay_pin, GPIO.OUT)
```

← Set up the PIN

```
GPIO.output(relay_pin,0) print("On")
```

← Send 0 through the pin

```
time.sleep(10)
```

← Wait function

```
GPIO.output(relay_pin,1) print("Off")
```

← Send 1 through the pin

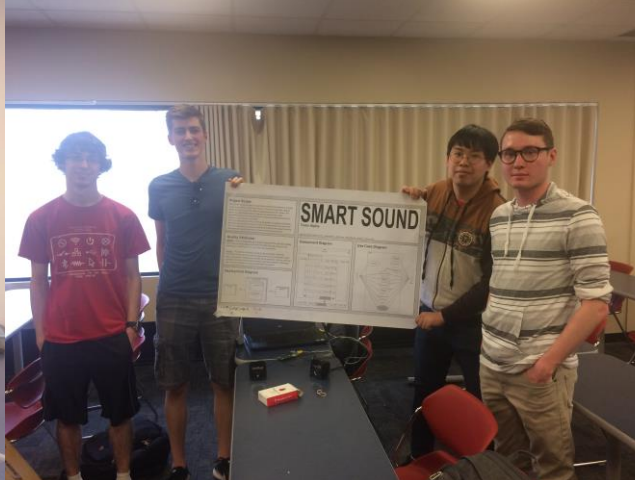
Try it -- Glowing Led

Reference :

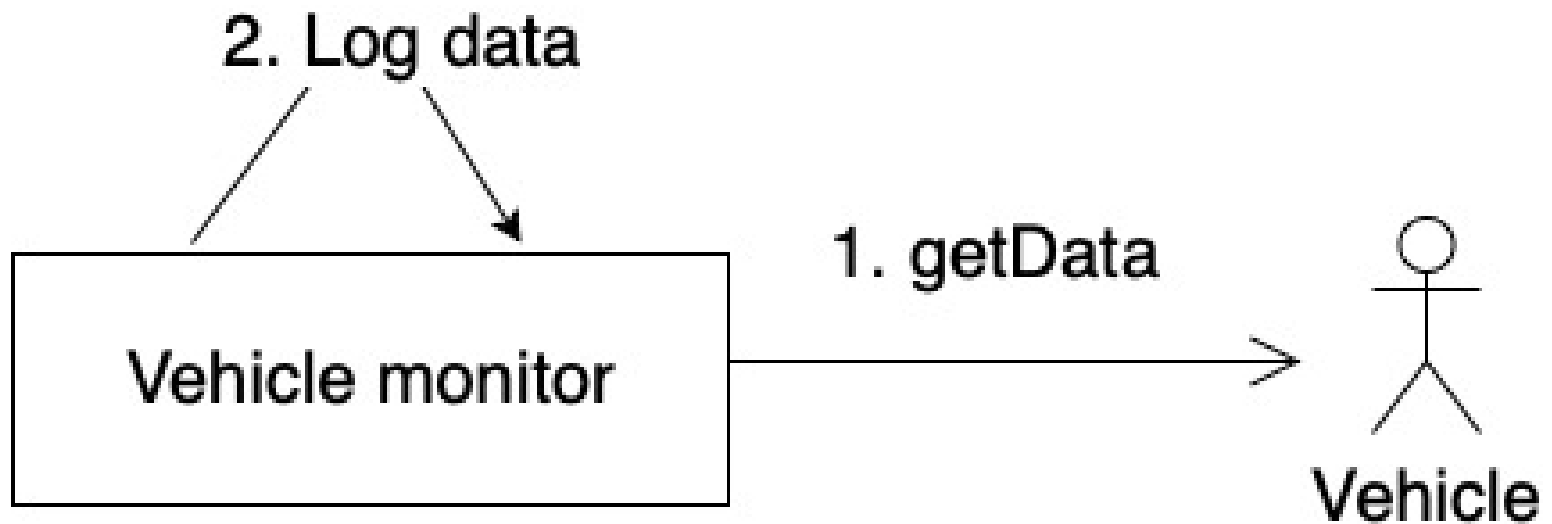
<https://www.youtube.com/watch?v=BWYy3qZ315U>

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
GPIO.output(18,True)
```

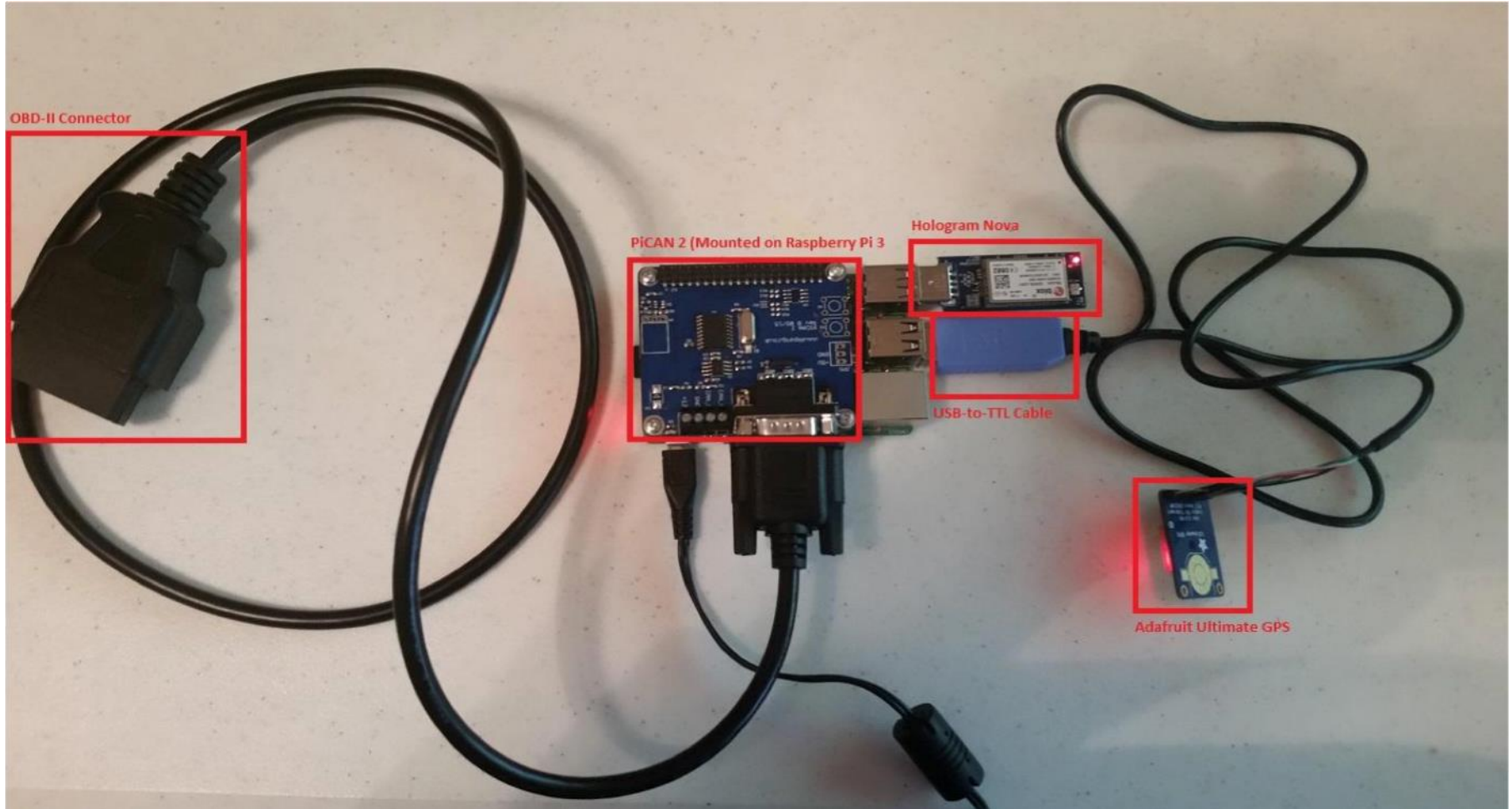
Using PI for Smart Home Projects



Fleet Management System



Fleet Management System



Basic Commands for Can Bus

```
import can
PID_REQUEST = 0x7DF
PID_RESPONSE = 0x7E8

bus = can.interface.Bus(channel='can0', bustype='socketcan_native')

message = can.Message(arbitration_id=PID_REQUEST,
data=[ID_FIRST_BYTE,
ID_SECOND_BYTE, pid, 0x00, 0x00, 0x00, 0x00, 0x00], extended_id=False)

bus.send(message)
message = bus.recv()

if message.arbitration_id == PID_RESPONSE:
```


Thank you

Next topic: Architecture for Distributed Systems