Azizbek 201939234.           Tolibjon 201939233A machine on the white cover

Description automatically generated with low confidence

**Abstract**

**Introduction**

**Background, used approaches (related work) and your project**

**System model**

**Block diagram**

**Implementation environment**

**Results**

**Conclusion.**

**Abstract.**

In this project we created a robotic toy car, which has several functions and can be used for different purposes. Such as delivering medicine to the patients in the hospital, checking their health through live video, and face recognition, motion detecting. Purpose of this project, since we had a pandemic, we all had to keep social distance, so it helps to doctors to keep in touch with patients repeatedly. IoT (Internet of Things) is now not only a concept, and It has end up not unusual place as character customers have tried to enforce it immediately via small forums and components. These studies carried out a machine to control a trendy toy automobile the use of Raspberry pi and motor driver. These studies will assist expand regular toy automobiles to RC automobiles or study on their very own and force on their very own via similarly development.

**Introduction**

Raspberry Pi Toy Car mission become evolved for the reason of studying how self-sufficient and self-using vehicles have been evolved, designed and the way without a doubt they work. This mission consists of Power Module motive force, Motor motive force module, a Robot Hat, Wide USB Camera, wheels and tires, and different vital elements for constructing the frame of the car. The relaxation of the features and different portions of system might be defined in element withinside the frame of the report.

**Background, used approaches(related work) and your project**

* A tool box
* Pi camera
* Raspberry pi
* Xiaorgeek car with 4 engines
* L298n h-bridge motor drive
* 6 batteries
* Keyboard
* Mouse
* Monitor
* HDMI cable

**Graphical user interface

Description automatically generated**

Basically, the Raspberry Pi runs the Raspbian OS, so we have to install the OS images on a microSD card. To do so, we need an SD adapter so a microSD card can be inserted into a regular laptop/computer's SD port (through which the images will be installed). Then, the microSD card is inserted into the Pi.

A picture containing text, electronics

Description automatically generatedGraphical user interface, website

Description automatically generated

We need to be able to work on the Raspberry Pi. Raspberry Pi is a single-board computer, so it can directly connect to a monitor. Be sure to connect the HDMI (using an adapter), USB (we’ll likely need a USB hub), and power ports. We’ll also need a keyboard and mouse (which you’ll connect using USB).

we can refer to the image to become familiar with the Raspberry Pi desktop. We can see a manila folder icon, which is the equivalent of a Finder (Mac). Essentially, it a visual of our files, downloads, directories, etc. For our purposes, it's how we'll manage our code. We'll also see a black screen icon, which is the terminal window. This is where we will write your code.

Open up a terminal window (there should be an icon in the upper left section of the screen).

* Run sudo raspi-config to get a menu like the one above
* Enable the P6 serial port
* Enable the P5 I2C
* From here, we have two options:

1. Continue using a monitor

* Every time we want to work on the Pi, just we have to do the same thing as above
* Connect the peripherals and work directly on the pi

2. Run the pi headless (using ssh)

* [https://www.raspberrypi.org/documentation/remote-a...](https://www.raspberrypi.org/documentation/remote-access/ssh/)
* Essentially, we can log in to the pi remotely from another machine to avoid having to connect peripherals
* To do this, we MUST enable ssh on the pi by following the instructions at the link above

Create a repository for the project. Then, in the terminal window, make a new directory for the project on the pi. We can edit our files within the terminal window (vi) if more comfortable, or using a GUI if less comfortable.

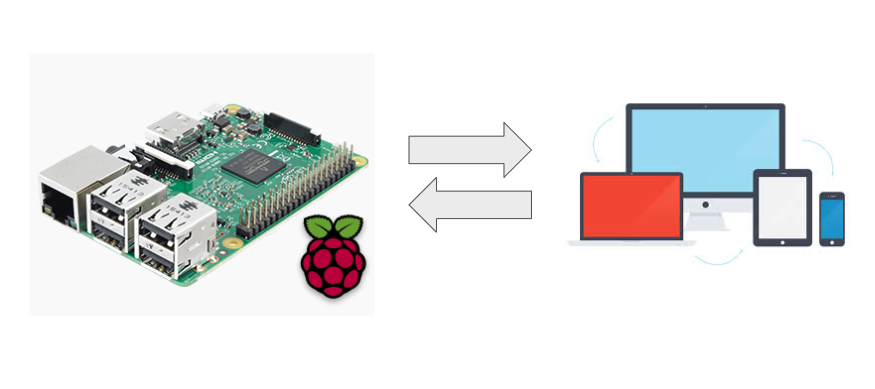
If you are unfamiliar with the command line, refer to these links:

[https://product.hubspot.com/blog/git-and-github-tu...](https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners)

[https://www.codecademy.com/learn/learn-the-command...](https://www.codecademy.com/learn/learn-the-command-line/modules/learn-the-command-line-navigation)

or do a Google search.

## Step 3: Prereading

[](https://content.instructables.com/ORIG/FXQ/LGX6/KM963UCX/FXQLGX6KM963UCX.png?auto=webp&frame=1&fit=bounds&md=4273a6142c6faf42dd5392b78394d3b3)

If we want to learn more about how all this stuff works (which I recommend), I would do some reading (there are plenty of resources and videos online!). But, here's a brief overview of what to know if we are just looking to get started.

* The Raspberry Pi board - a single-board computer; know where these things are (see image above):
  + MicroSD card slot
  + Mini HDMI port
  + Micro USB port
  + Micropower port & LED that indicates that the Pi is being powered
  + Camera connector
  + Processor (and heat sink if you have one)
  + 40-pin header & GPIO pins
    - Know the SDA/SCL pins, MOSI/MISO pins, and TX/RX pins
* Communication protocols
  + UART: universal asynchronous reception and transmission
  + I2C: inter-integrated circuit
  + SPI: serial peripheral interface

A close-up of a car engine

Description automatically generated with medium confidence**Results**

A toy vehicle on a table

Description automatically generated with low confidence

A picture containing indoor, yellow, cluttered

Description automatically generated A picture containing text, indoor, wall, electronics

Description automatically generated

Conclusion

These instructions were high-level because a lot of the fun is the building, trial-and-error, and eventual success! Hopefully, you enjoyed the building process. To work the car, you can use the four arrow keys to get the car to move around. Open a new tab and go to the IP address to see the GPS and compass data as well as the camera footage.