13/8/2024

TEAM 3

DAY 2 REPORT

DONE BY:

Ezzeldin Fekry

Nour Rizk

Nour Zeidan

Omar Hafez

Omnia Farouk

Today's Progress

1. Roles of Team Members

1.1 Hardware Sub-team

Ezzeldin Fekry Abdelsalam

- Implemented the footprints of the components shown in Figure (1).
- Conducted a thorough review of the schematic and made necessary modifications to certain components in the circuit.
- Searched on Arduino Bluetooth app to assist Omar tomorrow to test his codes.

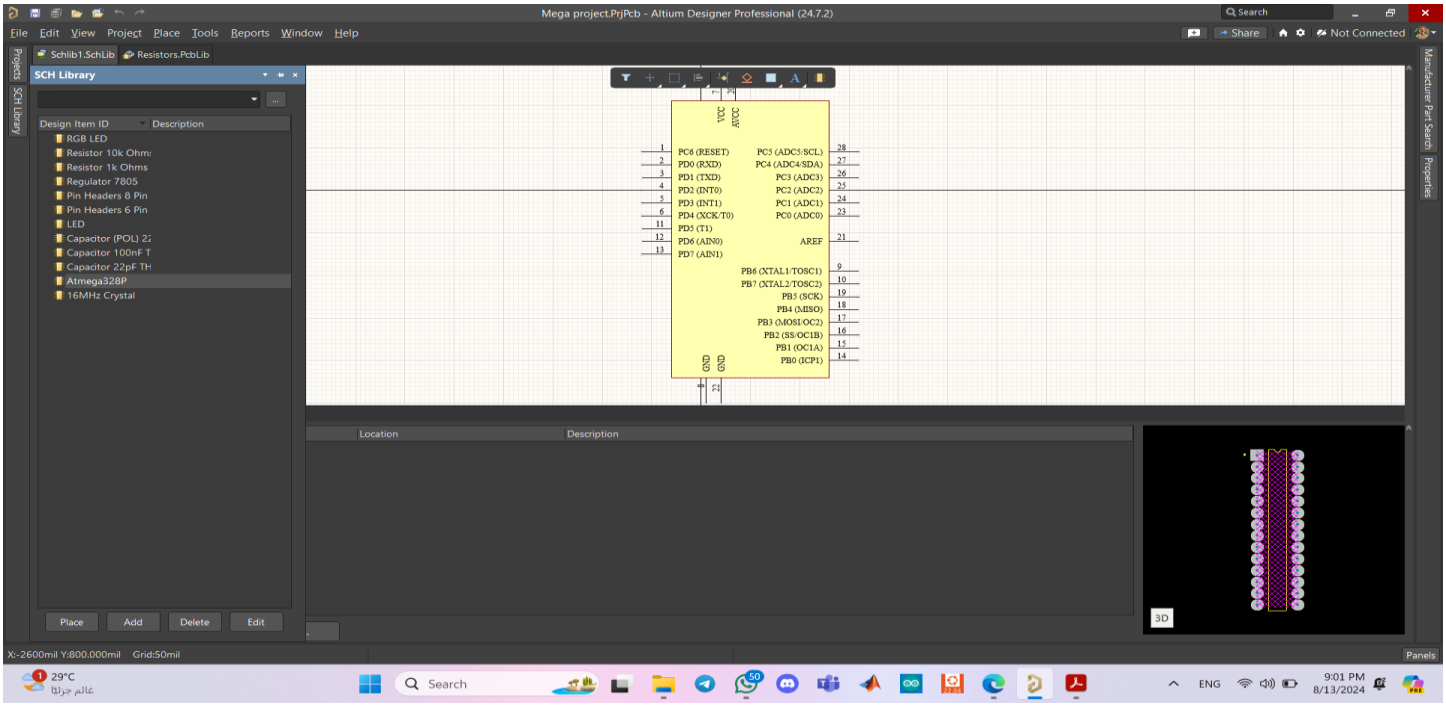


Figure (1) List of the implemented components

Nour Mohamed Ramadan

Today I searched for other components and got the details of each of them.

<u>A Decoupling Capacitor</u> (bypass capacitor):

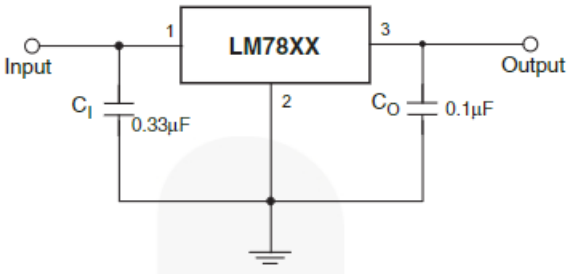
It is a capacitor used to smooth out voltage supply fluctuations and noise, providing a stable voltage to an active device such as a microcontroller or an IC. It is like a bodyguard which makes sure everything runs smoothly and if there is any noise in the power, the capacitor absorbs it.

<u>Suggested Values:</u>

1. 0.1uF (100nF) Ceramic Capacitor: is excellent for filtering out high frequency noise. You should place it as much as possible to the Vcc and the GND pins of the ATmega328p.
2. 10uF Ceramic or Electrolytic Capacitor: this value helps with lower frequency noise and stabilizes the power supply, especially if there are sudden changes in current demand (like turning an LED or motor on and off).

The best choice is to put the two capacitors together which provides more stability and noise filtering. But if we choose only one capacitor the choice goes to 0.1uF because ATmega328p operates at high frequencies up to 20 MHZ and this capacitor works with this range of frequencies filtering out the noise and the fluctuations.

Regulator 7805:



It is a linear voltage regulator that outputs a stable 5 DC from a higher input voltage.

-Output Voltage: 5V DC (±2% tolerance)

-Input Voltage Range: 7V to 35V (7V to 25V is more common for stable operation)

-Maximum Output Current: 1.5A (with proper heat sinking)

-**Operating Temperature**: 0°C to 125°C.

It is recommended to use a 0.1uF ceramic capacitor close to the output pin and a 0.33uF capacitor on the input.

Heat Sink :

The LM7805's internal components (mainly the pass transistor) convert the excess energy into heat. If not effectively managed, this heat can cause the regulator to overheat, potentially leading to thermal shutdown or permanent damage. Also, the regulator includes a thermal shutdown feature that disables the regulator if it gets too hot. While this protects the regulator, it stops the circuit from functioning until the regulator cools down.so we need a heat sink. And we choose the aluminum heatsink u-shape for TO-220 transistor.

How a Heatsink Works: A heatsink increases the surface area available for heat dissipation. This allows more heat to be transferred from the regulator to the surrounding air. And are made of materials with high thermal conductivity, like aluminum or copper, which efficiently transfer heat away from the regulator.

The effectiveness of a heatsink is measured by its thermal resistance (°C/W), which indicates how effectively it transfers heat. Lower thermal resistance means better heat dissipation.

1.2 Firmware Sub-team

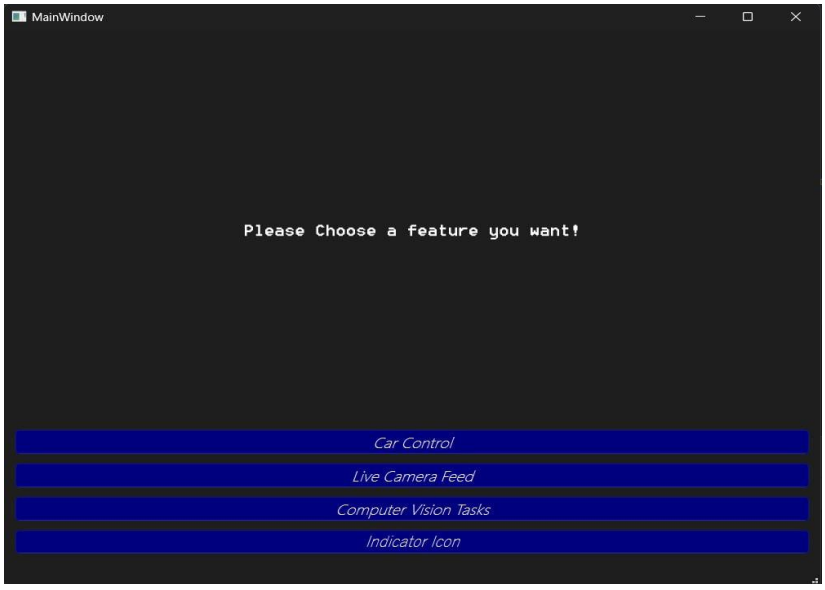
Omar Mohamed Hafez

- Finished all the basic operations and functions to move the car and control it and added the sensors function and added some extra features for the autonomous motion ; all the progress and the code will be found on Tinker Cad on this Link: (https://www.tinkercad.com/things/24p6XnmEoFc-project)
- Searched on how to use the Arduino as an ISP to run the code to Atmega328p and will be tested tomorrow after getting the components.
- Searched on how to use Bluetooth module and also will be tested tomorrow .

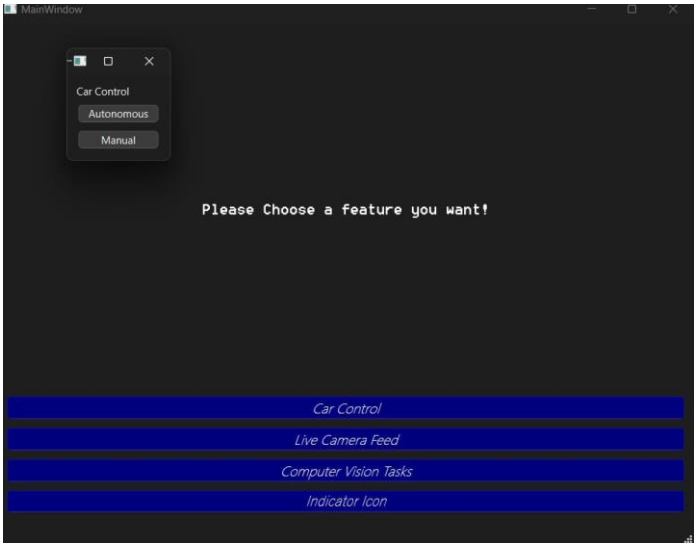
1.3 Software Sub-team

Nour Zeidan

Today I changed the main window's screenshot button to computer vision tasks as shown below.

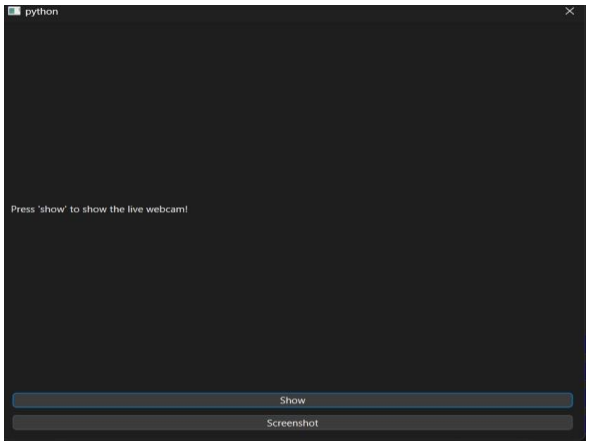


Secondly, I added a main window in which when car control push button is clicked it opens a new window containing two push buttons autonomous and manual as shown below.

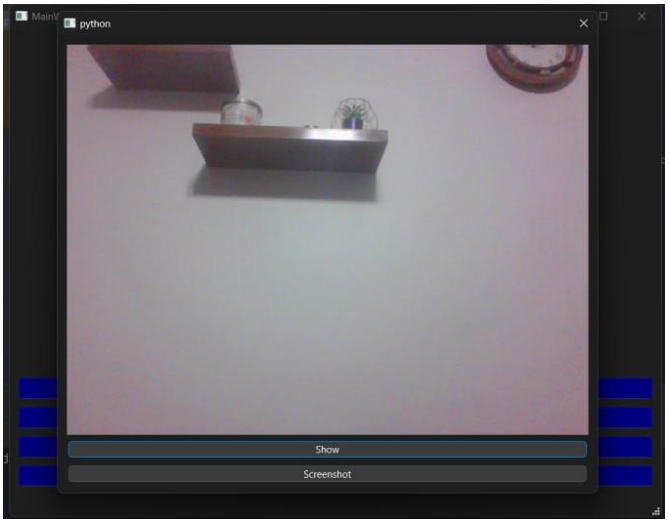


Lastly, I put a dialogue for the live camera feed button containing two push buttons first is ‘show’ to show the live camera and the other one is ‘screenshot’ to take a screenshot of the live camera as shown below, opencv was imported and used.

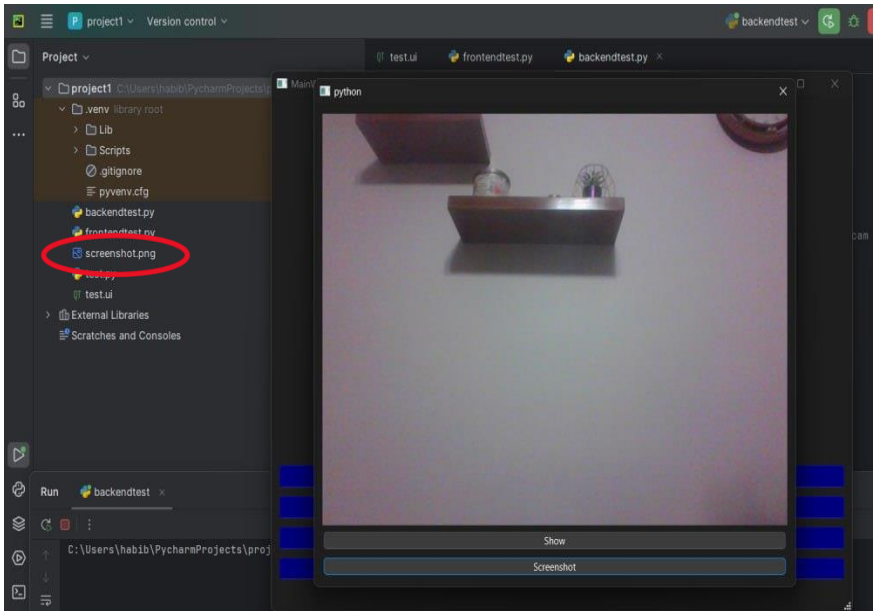
Dialogue:



Live camera when show is clicked:



When screenshot is clicked:



Omnia Farouk

➢ <u>Points to be done in the process</u>

The algorithm of the task must be done as stitching images, so for each frame processed from the left and right view videos, the images are stitched and displayed as one larger image in another frame until the video ends.

➢ <u>Video Stitching Algorithm progress</u>

- Finished Displaying the 2 frames of the left and right videos but not the GUI yet
- Fixed all path problems that occurred while reading video paths from File Explorer
- Used ‘Pathlib’ for the path-fixing process
- The video is displayed in a loop until the user quit and the program terminates
- Used the ‘q’ key for closing the display instead of the close button