PROJECT ALTAIR

Week 1

Hasin Mahtab Alvee
210042174
SWE'21

Contents

1 Theoretical Part

1.1 Task 01

Theoretical 1. Architecture for a self-driving delivery robot

While a SBC is a full computer with a RAM that can run a complete OS and run all other tasks similar to a computer, the micro-controller can only run specific commands as instructed.

For autonomous driving of a robot, the SBC is a rather safer choice as it can do multiple things at a time, but the micro-controller is a much more efficient choice when it comes to a smaller scale project.

For this scenario, The robot has gear motors and PWM-controlled motor drivers for wheels, servo motors for steering, a rotary encoder, GPS, IMU, and a Stereo Camera. All of these can be brought together with a simple micro-controller, for example - **Arduino UNO**.

So here, the brain will be an Arduino, which can communicate with all the other components. The GPS will start by getting the current location and the target location to set a track for the car. The Motors will be controlled by the PWM motor drivers. These drivers will send the motors to be turned on or off based on the current track obstacle. If there is no obstacle the motors will run on full speed. And to determine the obstacles, we can use the stereo camera. It can sense the obstacles to it's right sensor or left sensor.

- If there is an obstacle to it's right - near, it can slow down, reverse to right and move to left.
- If there is an obstacle to it's left - near, it can slow down, reverse to left and move to right.
- If there's an obstacle to the right - far, it can start to move left.
- If there's an obstacle to the left - far, it can start to move right.
- If there are obstacles to both side - far slow down and stop when they are near.

A rotatory encoder can take care of the turning left and right with inputs from the sensor data and then the IMU can keep the car from losing it's balance while turning or reversing from the data by slowing down the motors or speeding them up.

The reason to pick an Micro-controller over the SBC is for more efficient alogorithms and less complexity. For a bigger scale project the SBC would be safer choice.

References

1. SBC vs Micro-Controllers
2. Functional Architecture for Autonomous Driving
3. Realization of Self Driving Vehicle with Microcontroller

1.2 Task 02

Theoretical 2. Protocol to establish communication between Arduino Mega and Nano

Asynchronous Transmission: Data is sent in form of byte or character. It depends of the amount of time needed to be specified for per byte, which is generally specified as 9600bauds/sec. This transmission is the half-duplex type transmission. In this transmission start bits and stop bits are added with data. It does not require synchronization. Such as UART Communication.

Synchronous Transmission: Data is sent in form of blocks or frames. This transmission is the full-duplex type. Between sender and receiver, synchronization is compulsory. Such as I2C and SPI Communications.

Synchronous communication is more efficient and more reliable than asynchronous transmission to transfer a large amount of data. It is also slower than Synchronous communication. Even though, the UART communication is very much simpler to connect both NANO and MEGA on a PCB, it still falls far behind synchronous communications when it comes to transferring large ammounts of data over limited time.

Between the two types of Synchronous communication methods, the I2C and the SPI, The SPI being a Full-Duplex connection, it can handle data travelling from both the master and the slave at the same time. It is doesn't require a slave id like the I2C, which also makes it so much more faster and can transfer more data. The SPI can work with multiple devices at the same time with the same clock, MISO, MOSI connections. Even though the SPI is only capable of keeping a strong connection within the range of 20cm, whereas the I2C can maintain even at a 1 meters distance, this is where the PCB comes in handy.

Connecting the Arduino NANO and MEGA onto a PCB with a SPI communication is viable as it does not require much distance between the two micro-controllers. As both NANO and MEGA have support for SPI, and has the SPI library to simplify the use, it is a very viable option to use SPI connection for connecting the two devices.

References

1. Synchronous vs Asynchronous Communication
2. Arduino UNO vs NANO vs MEGA
3. Arduino UART Communication
4. Arduino SPI Communication

1.3 Task 03

Theoretical 3.

1.3.1 A.

Stepper Motors convert a pulsed digital input signal into a discrete mechanical movement. It does not rotate in a continuous fashion and moves in discrete steps or increments. A Stepper Motor is particularly well suited to applications that require accurate positioning and repeatability with a fast response to starting, stopping, reversing, speed control, high holding torque, and lower acceleration. For example, The stepper motor mostly used in 3D printers, CNC machines, medical imagery machinery, printers, security cameras, and other precisely controlled applications.

DC motors have been around since the 1830s and so have been used in a variety of applications. The DC motor, due to its improved working characteristics is still best used in applications requiring constant torque across the motor's speed range. It is fast and continuous rotation motors mainly used for anything that needs to rotate at a high rotation per minute (RPM), and examples are; fans being used in computers for cooling or car wheels controlled by radio, phone vibrators, power tools, car windows, cranes, conveyors, and many more applications that prioritize both sustained output power and price.

The stepper motor is so easy to use, but require some form of a microcontroller to help synchronize their rotor from one pole to the next. The DC motor requires an input voltage to its two leads and does not require external inputs for operation.

1.3.2 B.

References

1. Difference between Stepper motors and DC motors
2. Arduino UNO vs NANO vs MEGA
3. Arduino UART Communication
4. Arduino SPI Communication

1.4 Task 04

Theoretical 4. Arduino Mega vs ESP32 vs STM32

The Arduino MEGA being the fastest in the Arduino family, but when it is compared to the other two at question, It is not as fast as it may seem.

In a simple test to iterate up to 1 million while also lighting a LED, the MEGA falls behind by a lot, it takes almost 7200ms. Whereas, the STM32 and ESP32 both respectively took around 960ms and 160ms. So now let's compare between these two of which can be best used to utilize all the requirements for the projects.

ESP32 being a dual core, is much faster than the STM32, and it has Motor PWM for controlling the multiple motors in any project. ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. With it's Hybrid Wi-Fi and Bluetooth Chip, it can communicate upto 200 meters in range. For a remote control drone controller it has a Infrared remote controller (TX/RX, up to 8 channels). It has Hall effect sensor, which can be used for the home automation. Finally the Ultra-low-power (ULP) co-processor can use power efficiently.

Which means choosing a ESP32 is better than a STM32.

References

1. ESP32 vs STM32

2. ESP32

3. STM32

4. Arduino MEGA

1.5 Task 05

Theoretical 5. Sensors for a Martian food delivery robot

As sensors play a vital role in any automation driving models, the Mars food delivery car will also need to utilize all sorts of sensors for locationization and positioning.

The depth camera is a better choice for depth mapping, because that's not really what telephoto lenses are intended to accomplish. When you want to map a small segment of an overall object space, but most of the applications for depth mapping — such as autonomous vehicles, as one example.

A 6 Axis IMU is built on top of a Gyroscope and Accelerometer. 9 axis may sometimes have magnetometer. IMU measures and reports a body's specific force, angular rate, and sometimes the orientation of the body using the before mentioned components. So to know the state of the robot on the uneven ground of mars, this is a must.

A LIDAR, Light detection and ranging - is also very important as it measures the distance from a specific object by sending light and catching the reflects in the sensor to measure the distance from the object.

An Odometry system will help the robots to move along the absolute positioning of the field. It uses position wheels which are implemented using optical shaft encoder, which keep the robot from fixing the errors but the down side being it needs to do complex calculation in a short amount of time, but using the ESP32 may solve that problem.

A sonar sensor can sense the movements of objects that might not be in the camera or depth camera sight. So to evade incoming collision from the blind spots, a sonar can be really helpful.

References

1. Depth Camera

2. IMU

1.6 Task 06

Theoretical 6. Sensors for a Martian food delivery robot

References

1. Depth Camera
2. IMU

2 Logical Part

2.1 Task 01

Logical 1. Architecture for a self-driving delivery robot

References

1. SBC vs Micro-Controllers

2. Functional Architecture for Autonomous Driving

3. Realization of Self Driving Vehicle with Microcontroller

2.2 Task 02

Logical 2. Architecture for a self-driving delivery robot

References

1. SBC vs Micro-Controllers

2. Functional Architecture for Autonomous Driving

3. Realization of Self Driving Vehicle with Microcontroller

3 Microcontroller Part

3.1 Task 01

Practical 1. Architecture for a self-driving delivery robot

References

1. SBC vs Micro-Controllers

2. Functional Architecture for Autonomous Driving

3. Realization of Self Driving Vehicle with Microcontroller

3.2 Task 02

Practical 2. Architecture for a self-driving delivery robot

References

1. SBC vs Micro-Controllers

2. Functional Architecture for Autonomous Driving

3. Realization of Self Driving Vehicle with Microcontroller