3-D Mapping Using Lidar sensor

Vishnu Parammal, Prithvi Malhotra, Harshvardhan Kanthode, Bharat Shetty

Mechanical Department

Society of Robotics and Automation (SRA),VJTI, Mumbai-400019

Contact

[Vishnu.130599@gmail.com](mailto:Vishnu.130599@gmail.com) : 9623135325

[Mprithvi08@gmail.com](mailto:Mprithvi08@gmail.com) : 9769925844

[Harshvardhankanthode99@gmail.com](mailto:Harshvardhankanthode99@gmail.com) : 9619890144

[Bshetty0505@gmail.com](mailto:Bshetty0505@gmail.com) : 9323034368

INDEX

Abstract 3

Introduction 4

Technical Background (Hardware) 4

Technical Background (Software) 8

Problems and their proposed solution 9

Result 10

References 10

ABSTRACT

The notion of generating a precise map of a terrain has been proven to be an invaluable addition to aviation technology companies in the recent years.

Ordinary maps are inconsistent in their evaluation of erect constructions and would have to be regularly revised with the rapid growth of infrastructure. 3d maps not only orchestrate the intricate details of an area, but also build on 2d maps by adding depth of perception and colour coding.

The aim of the project was to contrive an interface between a light sensor, a microcontroller and a computer, design a software to effectively and accurately plot the entirety of an enclosed space on a 3-dimensional traversable map.

The selection of the appropriate sensor was done considering the cost, range, ease of programming the interface and the reflection complications.

We constructed a mount for our light sensor which was driven by 2 motors, one for the panning movement and the other for the tilting mechanism. The point-cloud mapping was implemented using a customised software.

The benefits of 3-d Mapping are immense and it is increasingly being implemented in cities for everything from Firefighting, Healthcare to Logistics.

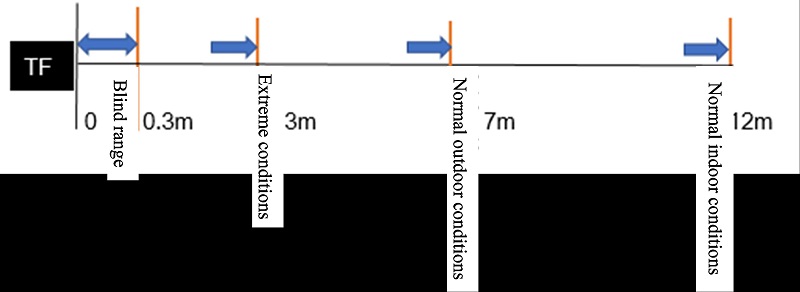
INTRODUCTION

HARDWARE:

* *Microcontroller*: Atmega32 is the high-performance, low-power Microchip 8-bit AVR RISC-based microcontroller which combines 32KB ISP flash memory with read-while-write capabilities, 1KB EEPROM, 2KB SRAM, 54/69 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for boundary-scan and on-chip debugging/programming, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a universal serial interface (USI) with start condition detector, an 8-channel 10-bit A/D converter, programmable watchdog timer with internal oscillator, SPI serial port, and five software selectable power saving modes. The device operates between 1.8-5.5 volts.



* *Lidar Sensor*: The TF-mini is a relatively cheaper Time of flight (ToF) light sensor which functions on the reflection of light off of opaque surfaces using a transmitter which sends out near-infrared light and a receiver, calculating the distance using the time difference or the phase difference of the light waves between the transmitter and the reciever. The range of the lidar sensor is upto 12 metres in normal indoor conditions and upto 3 metres in extreme conditions, with an error correction of 2%. The communication interface of the sensor is UART. Its operating voltage is 4.5-6 volts.



* *WiFi-Communication Development Board:*

The Wemos D1 Mini is a development board which has an in-built Esp8266 Wifi Microchip and a TCP/IP stack, capable of transmitting data sent from the Lidar to a localized server which the Laptop can access using a shared network. Originally the project involved sending Lidar data through a USB Type-A connection, and the Esp8266 was used to add the feature of portability to the mapping system.



* *Motors:*

Since we wanted to synchronize the movement of the motors with the acceptance of the Lidar data, the type of motor to be used was of paramount importance, and our finalized preference was using Stepper motors. Stepper motors work on the principle of magnetising multiple coils arranged in a ‘phase’ formation, by pulsing them in pre-determined intervals. The driving shaft of the stepper motor is rigidly attached to a disk with gear-teeth which get attracted towards the coils in intervals, thus ‘stepping’ the driving shaft of the motor. These motors grant an elevated control over the pan-and-tilt mechanism of our mount. The model we used was the NEMA-17 Bipolar Stepper with a 4.2 Kgcm Torque.



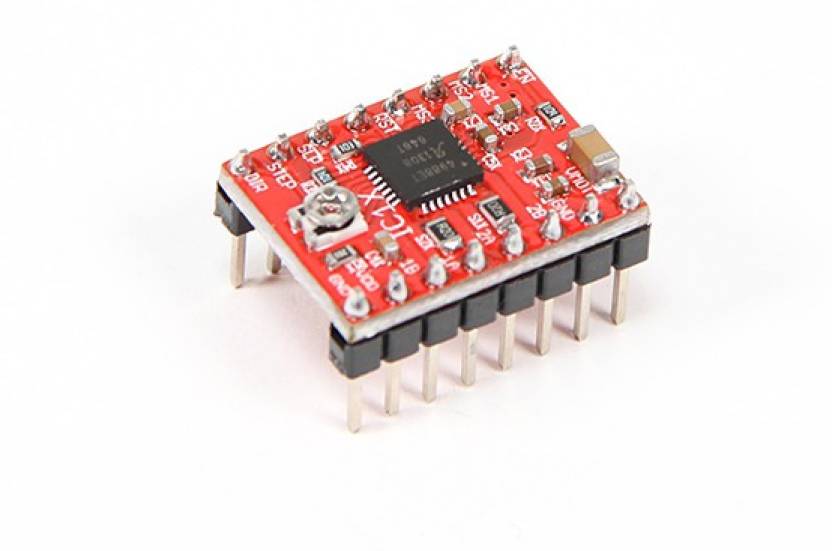
Nevertheless, such motors are difficult to control, as they require a specialised motor driver which needs adjustment according to the angle covered/ turn, by the shaft.

By trading control over accessibility, mid-way through the lifespan of the project, we decided to use Servo Motors instead, and they ran without any hiccups.

* *Motor Drivers*

There are numerous drivers that can run the NEMA-17 stepper, namely the A4988, DRV88xx motor drivers.

We used an A4988 break-out board, with 100mohm sense resistors, and microstepping modes available upto a 1/16th step, after adjusting the current using the on-board potentiometer.



SOFTWARE:

* *Arduino IDE:*

Arduino IDE is an open-source text editor used to communicate with AVR architecture microchips. The Atmega32, Atmega16 and the Wemos D1 Mini were programmed using Arduino IDE. The code for the Wemos was straightforward, the intention was to move both the steppers/servos and with each step/rotation, using nested loops the serial port would accept the sensor data using Serial.Available()

* *Python/PySerial:*

Pyserial was used to read the data sent by the Lidar through the serial port and interpret it in the form we needed for our mapping software. We also added formulae for converting the Polar coordinates into their Cartesian equivalent

* Matplotlib:

Matplotlib is a plotting library for Python. We used the Pyplot module along with the Pyserial code, to render the map. It is an object-oriented API, and has a similar interface to MATLAB. The mapping was possible in both real-time and after all the point-cloud data had been stored into a text file. It proved to be unintelligible and counter-intuitive to the purpose of the project.

* *OpenGL*:

OpenGL is a graphics Application Program Interface (API) that communicates directly with the graphics card drivers to render 2d as well as 3d vector graphics. Using C++, we implemented a software that reads the data sent by the arduino, analyses and converts to Cartesian coordinates and plots to the screen. Each point is created in the screen as a cube with its colour varying depending on distance and angles. The maps are scanned and plot in real-time, and it is possible to navigate the map while it is being created, though not advisable.

PROBLEMS ENCOUNTERED

* Controlling the Stepper- The most time-consuming obstacle in our project was to running the Stepper motors in the manner we envisioned. The motor drivers need to be precisely controlled in terms of the current that they pass to the stepper motors. Each version of the A4988 has different configurations of resistors and have to be identified to regulate their Reference Voltage. This was done by navigating to the official website of the motor drivers and using the formula provided on the data sheet, adjusting the potentiometer on the driver to limit the current sent to the NEMA-17 motors.
* Connecting the Lidar- The lidar sensor uses a unique 4-pin convertor that is rare to find. Initially the lidar was not transmitting, observed clearly due to the non-existent red light emitting from it. The problem was assumed to be in the convertor, however the problem was in the loose connections of the TX,RX,5V and Ground pins. It was resolved by soldering the wires to a 4-pin Relimate.
* Sending Data from the Lidar- The Arduino development routines for the Lidar available on the official webpage of the TF-mini were not accurate and caused the Lidar to send out garbage data. By modifying the data-types of the distance and strength variables, the Lidar started sending meaningful data.
* Data Format: The more popular Lidar sensors such as the Lidar Lite V3 collect data in a special .las format. There are many softwares available which are designed around the .las data format. The TF-mini collects data in a normal .txt format. The conversion problems in the format of data entailed us to collect the data in a string format and convert it to the format needed to map the points.

RESULT

The completed project constitutes of 2 stepper/servo motors which are can be observed to be moving in a covering approximately a hemisphere of space. The Lidar sensor mounted on an arm is mapping the data simultaneously. The point cloud data is comprehensible and can be navigated and toggled using the keyboard or a mouse.

References

* <http://www.instructables.com/id/Drive-a-Stepper-Motor-with-an-Arduino-and-a-A4988-/>
* [www.arduino.cc](http://www.arduino.cc)
* <https://forum.arduino.cc/index.php?topic=522948.0>
* <https://www.youtube.com/watch?v=89BHS9hfSUk>
* <https://www.youtube.com/watch?v=kJW9Uy4j-6k>
* <https://robu.in/product/tf-mini-lidar-laser-range-finder-security-li-dar-sensor-drones-uav-uas-robots/>