AC 2009-600: CONTROL SYSTEM PROJECT: RFID-BASED ACCESS HUMAN TRANSPORTER

Sriram Sridharan, Texas A&M University

Sriram Sridharan is EE PhD student. He works in the EET as a teaching assistant while pursuing his doctoral at Texas A&M.

Ben Zoghi, Texas A&M University

RFID-Based Access Human Transporter

Abstract

The undergraduate research paper documents the implementation and testing of a Segway-like scooter utilizing an MIT DIY Segway design, while adding passive RFID access. Additional features include an adjustable handle bar, lean steering using car struts, an IPOD holder and speakers. Our proposed approach is to base the overall design on the MIT DIY Segway. If a person steps onto the platform the scooter will have the capability to stay up-right. When the rider leans either forward or backward the scooter will accelerate in the correct direction to center itself. Steering will be controlled by leaning the handle bar to allow the rider to turn in the desired direction. The RFID reader will turn on the Transporter only when an authorized tag is placed near the antenna. This security feature will allow only an authorized user to ride the Transporter.

Introduction

The Transporter project will be utilizing the automatic identification method that makes Radio frequency identification (RFID) so unique. The components that enable RFID to be so useful are the exchange of data between a passive tag and reader. The passive tag requires no internal battery power but becomes active by being energized when a when a reader is near. A passive tag is powered is through radio waves that a reader outputs. For our purpose the read range is small so we do not require an active tag that can be read from farther away. Having the read range limited to a few centimeters protects the rider from accidentally turning the Transporter when the rider is not ready.

Using RFID for the power on switch for the Transporter our application allows for security and protection from harm. Since we are using passive tags our read range for our antenna is 3-5cm. This short distance allows the rider to turn the Transporter off and on only when the tag is close enough to be read. There is no danger of the tag being read from across the room and turning the Transporter off or on. On the security side, the Transporter will only be turned on when an authorized tag is place in the antenna field. A data base of authorized tags is stored on the RFID reader itself. Most vehicles like the Transporter will have a button or a key that toggles power on or off. A key or switch can be duplicated and overcome by an un-authorized person. That is why having keyless power-on is so unique. To overcome our security you would have to reprogram the reader and that takes time and requires several pieces of hardware. The Transporter database currently only allows two tags to have the authorization to turn it off or on.

The Transporter's general requirement is to balance a single person on a two-wheeled platform. The rider will be able to balance themselves on the platform while holding a handle bar located directly in front of them for support. The motors will take in a PWM signal that is outputted by a PIC16F877 processor. The PIC runs a 4MHz clock that will poll the sensors at 100Hz. There are two sensors that allow a person to stay upright. Those sensors include two accelerometers and a single gyroscope. One of the accelerometers monitors acceleration. Measuring the acceleration helps us to know when the Transporter is tilted in a certain direction and adjust the motors to keep the Transporter upright. The other accelerometer measures the tilt of the steering bar. The rider can tilt the handle bar in two different directions to turn in the desired direction. The gyroscope measures the angular rate or the speed of the rotation. All three of these sensors allow the rider to stay upright on the Transporters base plate and to travel in any desired direction. This paper outlines the implementation of the DIY Segway that was designed by MIT students with the added addition of RFID.

System Description

The Block diagram of the system is shown in Figure 1. Each component and the integration of each into the complete system will be detailed later in the paper.

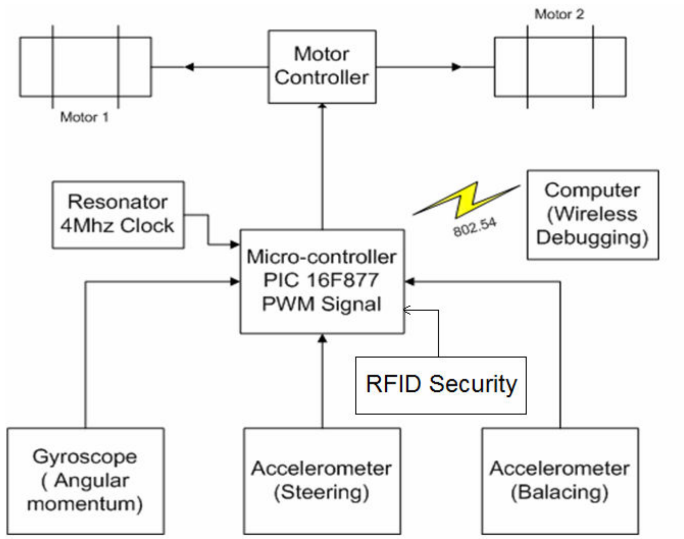

Figure 1.Block Diagram of Transporter

The system components can be broken down into two distinct categories: hardware and system software. The first group, hardware, consists of the Passive RFID equipment, motors, motor controllers, PCB, battery, structural pieces, and sensors. The software category includes the RFID reader software and the C program to monitor sensors and adjust the motors.

Figure 2.Transporter

Hardware

The first component in the hardware category is the passive RFID reader. This reader's function will be able to log and verify the presence of certain authorized tags and when they are read, to adjust a relay to power the motors on or off. The tags will be placed in identification cards so they are easy to carry around in your pocket.

I. Passive Reader

The Passive reader being used for this implementation is the Micro RWD MF (Mifare) ISO14443Reader Module. It transmits at 13.56 MHz and is capable of ranges of up to 3-5cm. The reason we used this reader is because of its limited read range and ability to store several different tags' serial numbers.

Power for the reader is supplied by a 12VDC motorcycle battery. We use a voltage regulator to decrease the voltage to 5V, which is what the reader needs to operate. The reader will be put on our PCB and an antenna will be run to where the rider will have easy access to it. Also on the reader there are two LEDs. One LED shows power from the battery while the other LED tells the rider that the Transporter is on. Once the reader detects a valid RFID tag, a J/K Flip-Flop reads the change and flips a low voltage relay which in turn flips a high voltage relay that powers the Transporter.

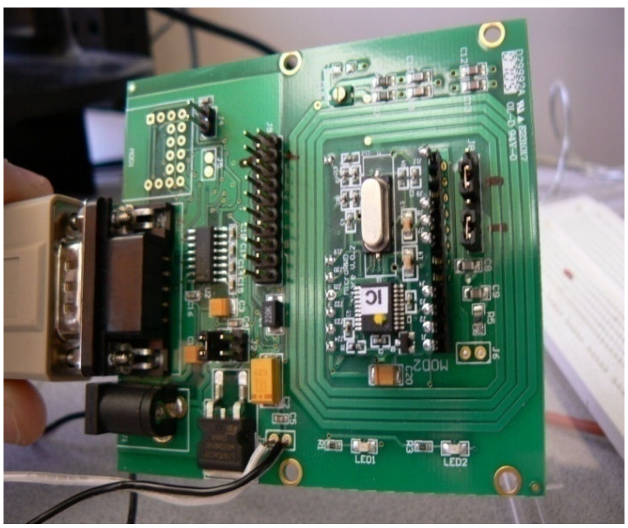

Figure 3.RFID Reader on the Development Board

II. Motors and gearbox

The FIRST CIM motors have the feature of a keyed output shaft. They require 12VDC and have a peak power of 3337W. The gearbox is designed by FIRST Robotics and has a 12:1 overall gear ratio reduction. Both of these parts together provide the torque to carry a single person. We lose a lot of speed because of this gear reduction, but speed is not the main objective. Instead, the torque needed and generated is the critical objective so that the motors will steady a person. The calculated top speed without a person is 12mph. Actual speed is far lower because it depends upon the weight of the person riding. Figure 4 shows the motors and gearboxes mounted to the base plate.

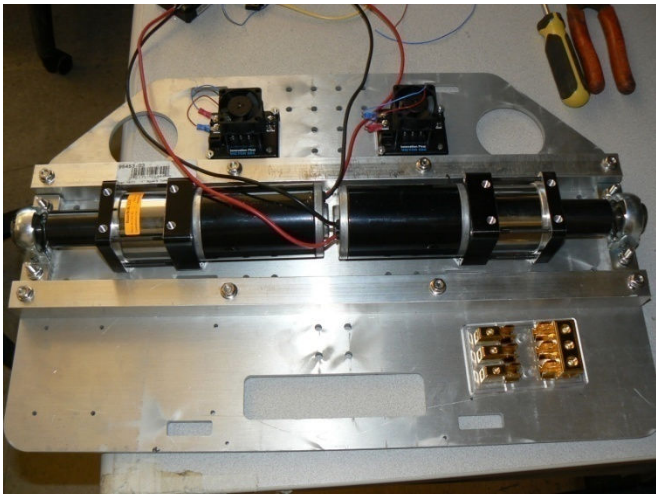

Figure 4.Motors and Gearbox

III. Base Plate and Fabrication

The base plate was designed in CAD by MIT students and cut on a water jet by Big Blue Saw, a metallurgy company. Aluminum ¼”-thick denoted by the code 6061, was used for the base plate. Several screw holes were cut into the aluminum plate to allow for easy attachment of several key parts such as the motors and each gearbox. Most of the support comes from 1”x1”x3’ aluminum box extrusion.

V. Couple, axel, and bearings

The couple allows the wheel shafts to be locked down safely into the couple’s receptacle. One shaft comes from the gearbox and the other comes from the wheel. They meet at the coupler and the key shaft is placed to aid the motor to turn the wheel. The axel is the shaft that runs from the gearbox to the wheel. The bearings give the axel freedom to turn the wheel in a smooth motion while giving the axel support near the end of the base plate.

VI. Wheels

The wheels are urethane tires that have off-road tread. We chose a larger wheel than the MIT students. The larger wheels allow more wear and tear on the tread over time because it takes less motion to adjust the wheels to stay steadily balanced.

VII. Handlebar

The handlebar is made from a piece of 1”x1” aluminum. The handlebar is designed to be adjustable. Four butterfly nuts hold the shaft, and each nut can be loosened to allow for the handle bar to be lowered or heightened and subsequently retightened for use.

VIII. Battery

We discovered fairly quickly during the testing phase that the Transporter will drain batteries rapidly. For example, a motorcycle battery gave 5-10 minutes of riding time. Also, while the battery was under heavy duress at the end of the battery's power, the Transporter would stall easily. We moved next to a car battery because it extends the Transporter's life span. Though the battery is sizable and weighty, it seems to actually aid the rider in balancing themselves given the distributed excess weight of the battery. Finally, a fuse block was used to protect the processor and various other components from being destroyed in a short.

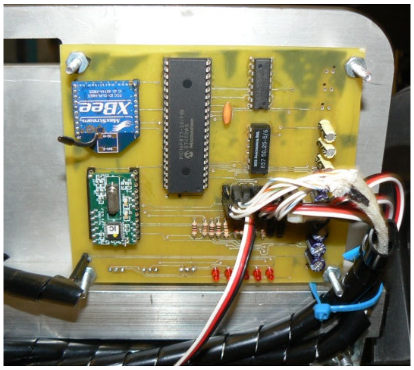

Figure 5.Control Board

IV. Sensors, Signals Electronics, and Controller

We are using three sensors total; 2 Accelerometers and 1 gyroscope. They both output voltage ranges from 0 to 5V. The neutral angle is 2.5V. We tested the sensors and added in a calculated offset to calibrate the sensors for proper operation. The accelerometer measures the acceleration of gravity in a certain direction. The gyroscope measures angular acceleration. In our case, we use the gyroscope to measure the inclination of the rider. Using both sensors allows us to find the angle to horizontal, or level in other terms. The last accelerometer is the steering for the Transporter. Since there is noise in the system, capacitors were placed on the battery power inputs. Each of the sensors and the voltage regulator will have capacitors to clean up the signal as well as to protect the different components against system voltage and current spikes that could cause harm.

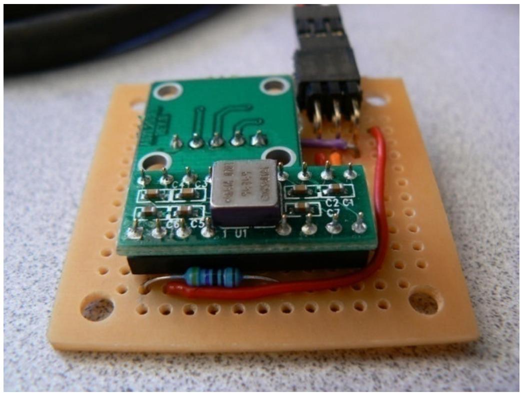

Figure 6.Accelerometer and Gyroscope

X. Speakers and IPOD Connection

Another piece of hardware that is unique to the Texas A&M University Transporter are the speakers and IPOD 3.5mm jack. The speakers are located on the base. The original holes were used by MIT students as cup holders. We thought that it would be too difficult to have functional cup holders that low on the Transporter so we put the holes to better use. We also mounted an IPOD case to the handlebar with a 3.5mm head phone jack ready to plug into any mp3 player.

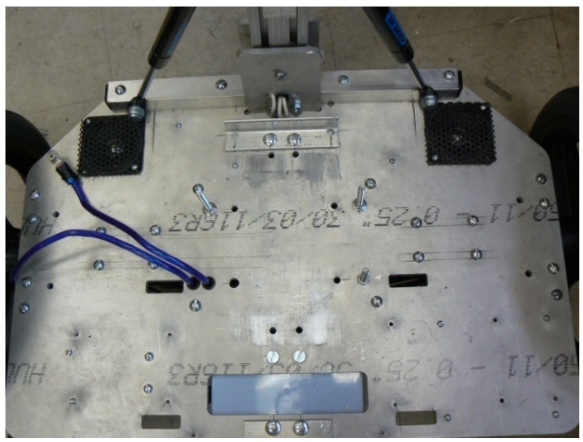

Figure 7.Speakers in base plate (top right and left black squares)

Software

MIT students designed a software program that enabled the Transporter to work as efficiently as possible. The use of this “premade” code allowed us to keep useful work time to a maximum while minimizing downtime. For programming and debugging the Transporter code on the PIC processor, Xbee[ZigBee] was used for wireless data transfer communication. The use of the Xbee gives added value to this system and is useful for monitoring the sensors from a distance with the additional advantage of no wires need be connected. The debugger used for the Transporter was the Machine Science’s compiler. Since the Machine Science programmer has created useful header files for the PIC16F877, the programming and debugging capabilities are very useful.

I. Algorithm

The program that MIT developed begins with the header files they included from the Machine Science website. These include files that have ADC and PIC as headers. These two headers setup the PIC for programming and taking inputs from the analog or digital sensors. After the initializations, the program sends out the data to the GUI program in Visual Studio that allows the user to monitor the output of the sensors as well as see the orientation of the Transporter. The Xbee is used to send out the data to a laptop that is running the GUI program. After the GUI program receives the data sent, the sensor pins are read and stored. After the different sensors are read, each sensor requires adjustments and calibrations to take into consideration offsets for correct scaling. The offset is actually a number that is realatively easy to calculate and fin because locating the neutral point on the sensor and reading the input of the sensor will give you the offset. Subtracting the sensor's neutral reading from itself will tell the program where the center of gravity for the Transporter is and where the balancing point should be. The scaling depends on the sensor that is being considered. Scaling is the factor by which we multiply to get the desired units. This type of information can be found on the sensor data sheet.

The next phase we need to talk about is the digital filter that is written as an algorithm inside the program code. In order to control the Transporter we need to know the angle, or pitch, and the angular velocity of the platform. In this instance, the MIT students designed a simple PD (Proportional/Derivative) controller, without the Integral function because the "I" in a PID controller requires a large amount of processing power. This part is simply not needed for accurate and efficient results of the Transporter's ability to move. The equation for the motor output, controlled by a Victor motor controller, is shown below:

Motor Output = K_P *x (Angle)* + K_D *x (Angular Velocity)*

The motor output can be controlled by K_P and $K_D$to give stability to the system. The controller enables the motor to smoothly and quickly transition the motors in the forward or reverse motions, to enable the platform to balance. If the filter were just a proportional controller, the change in overshoot would be too high and would make the Transporter wobbly and really very awkward to ride. That being said, the availability and implementation of a PD controller, to show the difference, allows the motors to adjust quickly and yet smoothly at the same time to avoid overshoot in the controller equation, and alleviates the probability of the rider getting hurt.

II. Graphical User Interface (GUI)

The GUI was also created by MIT students to allow for real time viewing of sensor data via Xbee. The ability to view the sensor data through this graphical user interface provides many advantages for the design and building phase of the Transporter. One advantage allows for easy offset discovery while the sensors are running and outputting data. Also included in the GUI is clock speed of the PIC processor, a simple 3D orientation guide, the power level of the motors and which mode of operation they are in, forward or reverse, and the current offset for the sensors to be read in correctly and calculated for efficient output of the hardware involved in moving the Transporter. Seeing each of these variables through a wireless communication device prompts the allowance of easy debugging as well as unique way to experiment with different offsets in the sensors to calibrate the most efficient riding experience.

Conclusion

In conclusion, the Transporter was a fun and technically challenging project. Even though our overall design was based off of MIT, we ran into many technical challenges that were not documented nor handed to us in any specification sheet. Through the combined use of mechanical hardware and electronic equipment, the Transporter has become an experience that is very enjoyable, but also very encouraging as a learning environment. Many of the parts that were used to build the Transporter, were new products in our hands. Exchanging information over Xbee, using a PIC microprocessor, our specific use of relays and a J-K Flip-Flop made this project a very worthwhile expenditure of our time and resources to gain specific knowledge involving electronic equipment. The addition of mechanical hardware operating in-sync with the electronics gave us considerable ability to learn about hardware with a very hands-on approach.

Bibliography

1. http://web.mit.edu/first/segway/