DIGGING AND SEED SOWING ROBOT

Team 7

Abhishek Zanane - 113054003
Alok Dudwe - 123050080
Pankaj Kumar - 123050066
Vinod Kumar Ora - 123050059

1. Introduction

This project is about modeling a Seed Sowing and digging Robot which can be used to help in sowing and digging seeds in a green house arena at some specific distances as per the type of seed to be sown . The robot will move by following the black line between the troughs and will first dig hole on one side of the aisle in a pattern and drop the seed into the hole. The distance in inches between seeds row wise can be selected by user through Zigbee module.

Definitions

ZigBee is a specification for a suite of high level communication protocols using small, lowpower digital radios based on an IEEE 802 standard for personal area networks. Detailed information on zigbee can be found in appendix-a.

Wheel Encoders The wheel encoder is a sensor attached to a rotating object (such as a wheel or motor) to measure rotation. By measuring rotation your robot can do things such as determine displacement, velocity, acceleration, or the angle of a rotating sensor.

2. Problem Statement

To build a robot which will dig hole and sow seed in a greenhouse arena as specified by the user by following black line and then turns back on sensing all three white by the three sharp sensors and returns to the original location (again by sensing all three white by the three sharp sensors).

3. Requirements

Functional requirements

The robots uses sharp sensors which follows black line and moves straight. It stops at specified distances by the user and then activates the servo mechanism and digs a hole in the ground and then activates another servo controlled piston mechanism and drops a single seed in the hole and moves forward. It continues to repeat this in a straight line turns back by 180 degrees on sensing all three white by the three sensors. It calculates the travelled distance and returns to the starting point and stops by sensing all three black lines.

Non-functional Requirements

Accuracy in determining and transmitting the value of distance by the user
Less delay in transmitting and receiving the signals.
The bot working should be acurate and repeatable.

Hardware requirements:-
Wheel encoder
Zig-bee USB adaptor
Onboard battery support
Digital IR sensor
Fire-bird ATMEGA bot
Two Servo motor
Seed dropping mechanism
Three white line sensors

4. Implementation

4.1 C Code Description:

The most important and best used feature of the ATMEGA -is ISR (interrupt service routine). ATMEGA is the basic building block of the SPARK-V. So let's have a look how an ISR works.

An **interrupt handler**, also known as an **interrupt service routine** (**ISR**), is a callback subroutine in microcontroller firmware, operating system or device driver whose execution is triggered by the reception of an interrupt. Interrupt handlers have a multitude of functions, which vary based on the reason the interrupt was generated and the speed at which the interrupt handler completes its task. An interrupt handler is a low-level counterpart of event handlers. These handlers are initiated by either hardware interrupts or interrupt instructions in software, and are used for servicing hardware devices and transitions between protected modes of operation such as system calls.

We have used ISR to serve our real time need of data sending through zigbee protocol and count the shaft counts. Shaft count is used to perform all the basic operations like “TURNING BY SPECIFIED DEGREE” “MOOVING BY A SPECIFIED AMOUNT OF DISTANCE”.First the user specify distance with the help of Zigbee and bot will start moving forward, when it travel that amount of distance given by user then we perform digging and sowing operation with the help of servo’s that is one for digging and one for the moving piston to drop accurately one seed at a time. This whole thing moving, digging and sowing are in the infinite while loop, and the function of moving on the black line is contain one more thing if it get all three line sensor detects white color then it will turn approximately (180 degrees), and call home function in which our bot simply follow black line and came to our starting point to identify starting point we put a check point there that when all three sensor sense white color than it will stop there.

5. Testing Strategy and Data

We have implemented & tested the robots in the provided arena of the green house. For testing we have tried various inputs in inches like 4,5,6,7,8,9 (single digit values) as a distance between seeds. The robot was tested with battery and without battery on power supply for numerous times and results were recorded to observe the behavior for repeatability and predictability. We found that the robot performs its task consistently and is repeatable and predictable. We have also implemented the movement of robot and found the robot to be following straight line more precisely at medium speed. The robot was tested for power consumption and energy calculation was done for it considering the energy consumption for two motors, three sensors two servo motors fitted on the robot.

Operations	**7 inch**	**3 inch**
Moving opr	3.5 sec X 0.86 Amp	1.3 sec X 0.86 Amp
Dig Servo	**2.1 sec X 0.8 Amp**	**2.1 sec X 0.8 Amp**
Sow servo	2.9 sec X 1 Amp	2.9 sec X 1 Amp
Total	**7.59 Amp sec**	**5.69 Amp sec**

Energy for going upto end at 7 inch gap = 7.59 X 12 = 91.08 Amp sec
Energy for returning to starting point = 0.86 X 55 = 47.3 Amp sec
---

TOTAL

= 138.38 Amp sec

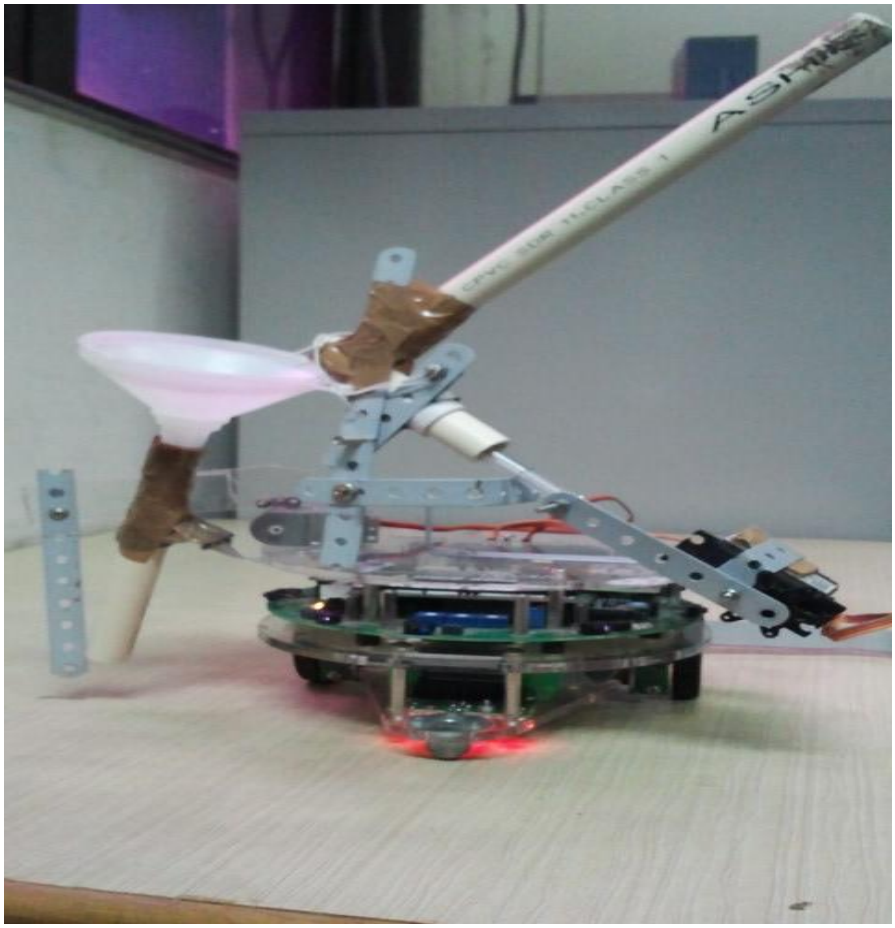
=38.44 mAmpHr

6. System description

The system consists of three parts :

In First module which helps robot to move forward, stop and returning back to starting point. This is achieved by programming for black line forward movement, stop function till the time robot operates digging and sowing tasks and again moving forward.

In Second module it helps robot in digging the arena at specified distances. In this the servo motor operates from x degree to y degree depending on the digging attachment and digs hole. Third module consists of seed sowing piston controlled servo mechanism. In this module after digging operation is complete then the single seeds will be dropped with the help of piston mechanism which allows only one seed to be dropped if you select correct degree of rotation of servo depending on the design of the attachment. The least requirements of these robots which are required for project are precisely working wheel encoders, motors, XBEE wireless, hole digging and seed dropping mechanisms. The details of implementation of the system has been described in implementation part.



Working

The robots is initially stationary and waits for user input through Zigbee module. User can give values from 4,5,6,7,8,9 as input .The bot treats these values as the distance in inches for distance between two consecutive seed to be sown. On receipt of input from user it moves forward upto that distance stops and perform hole digging operation with the help of arem available on the bot which is servo controlled. We have achieved this in the program by a servo function. Then after digging it drops exactly one seed from the nozzle with the help of piston mechanism which is also servo controlled. The robot continues to do the same moving forward, digging and sowing operation until it detects all three white by the three sensors. It then turns back by 180 degrees on sensing all three white by the three sensors. It calculates the travelled distance and returns to the starting point and stops by sensing all three black lines.

Things that worked as planned

The code has been written as per the requirements and it works fine and efficiently starting from giving distance by the user through Zigbee module and moving forward motion of the robot, and carrying out digging and sowing operations. Initially we had trouble in moving straight on black line, but it is after achieved by adjusting sensor intensity. We had also get stuck in how we can drop exactly one seen at a time , it is achieved by lots of hit and try models which are able to throw seed and finally we used a piston kind of mechanism which can drop exactly one seen at a time which is servo controlled.

Things that did not work as planned and their reason:

We have faced the problem in moving the robot exactly the same amount of distance provided by the user by an accuracy of +/- 1 cm. This problem is due to initially we were under the impression that the error in calculating the coordinates could be due to slip in wheels. But we started to face problem with the variation in velocity of the two motors. To overcome the above problem, we tried to calculate the difference in the velocity of the two motors and hard coded by reducing the velocity of one wheel with the difference calculated. This worked and solved the problem with the accuracy of +/-1 cm. We had planned to move the exact distance but could only achieve with accuracy of +/- 1 cm.

Simulation

We have simulated the robots in the provided arena of the green house. We have tried various inputs in inches like 4,5,6,7,8,9 (single digit values) as a distance between seeds. The robot was simulated with battery and without battery on power supply for numerous times and results were recorded to observe the behavior for repeatability and predictability. We found that the robot performs its task consistently and is repeatable and predictable. We have also simulated the movement of robot and found the robot to be following straight line more precisely at medium speed.

7. Future Work :

The most important aspects of a project is scalability along with desired efficiency. When we are dealing with this robot, scalability of project in terms of moving the bot for performing digging and sowing actions is of great importance. Our Project is having an efficient scalability which can be extended for very large size (large problem size means operating in a large arena).

It can also be used in certain military operations where human life risk is involved like digging and burring mines. These robots can be deployed for some specific household jobs also as it can traverse inside the house and as per the work modified attachment can be fitted house like for hold applications to clean the houses.

A camera can be mounted on the bot which will give the image (and if multiple cameras mounted views from multiple directions can be taken and processed). By mounting the camera on top of these robots, we can send these robots with recording features of the digging and sowing operations. These robot can also be utilized n this way for surveillance purpose purpose with modified movement algorithm as they can be fitted with camera and can be send in any type of arena to capture live video recording and then sending it to a person sitting in control room.

8.Conclusions:-

This project can be used for doing mechanized operations like digging sowing and plough arena in the greenhouse arena. It can also be used in certain military operations where human life risk is involved like digging and burring mines. These robots can be deployed for some specific household jobs also as it can traverse inside the house and as per the work modified attachment can be fitted house like for hold applications to clean the houses.

Further, these robots can also be used for surveillance purpose with modified movement algorithm as they can be fitted with camera and can be send in any type of arena to capture live video recording and then sending it to a person sitting in control room.

The things that seem to be achievable using software simulation is not always accomplished as expected due to dependencies on hardware behaviour. So the feasibility of the project (especially an embedded project) should be analyzed from a hardware perspective ,Real time analysis . Testing and Simulation is essential to predict the behavior in case of repeatability and predictability.

9. References :

[1] MATLAB R2011b Documentation [url : http://www.mathworks.in/help/techdoc/index.html]
[2] STAGE 4.0.0 Reference manual [url : http://rtv.github.com/Stage/]
[3] WinAVR User Manual – 20100110 [url : http://dybkowski.net/download/winavr-usermanual.html]

APPENDIX-A

XBee and XBee-PRO 802.15.4 OEM RF modules are embedded solutions providing wireless end-point connectivity to devices. These modules use the IEEE 802.15.4 networking protocol for fast point-to-multipoint or peer-to-peer networking. They are designed for high-throughput applications requiring low latency and predictable communication timing. While Bluetooth® focuses on connectivity between large packet user devices, such as laptops, phones, and major peripherals, ZigBee is designed to provide highly efficient connectivity between small packet devices. As a result of its simplified operations, which are one to two full orders of magnitude less complex than a comparable Bluetooth® device, pricing for ZigBee devices is extremely competitive, with full nodes available for a fraction of the cost of a Bluetooth node. ZigBee devices are actively limited to a through-rate of 250 Kbps, compared to Bluetooth's much
larger pipeline of 1Mbps, operating on the 2.4 GHz ISM band, which is available throughout most of the world.

What is X-CTU?
Digi International offers a convenient tool for Xbee module programming - X-CTU. With this software, the user be able to upgrade the firmware, update the parameters, perform communication testing easily. The basic operations are listed below.

How To Use X-CTU?
Before we can talk to an XBee, except USB cable we will need to get an USB adapter for XBee With the USB adapter, we can communicate with Xbee through **"USB Serial Port"**. We may have more than one device on serial port, for example, we want to test the wireless communication and connect two XBee to our PC. So, we will add more devices on serial port. In either case, we need to select the correct one that we want to perform operations.In this case, it only has one Xbee connect to PC and it locates on com port 9 (COM9).

Xbee Query:
This is an easy way to test and check if an XBee is working and configuring properly. After parameter modification and firmware upgrading , we need to do this for checking if everything is ok. we will have a very little chance to get a problem Xbee. If we got problem to Test / Query an XBee , usually it is due to the wrong parameter setting.
For a successful two way communication, the most primary principle is the **"Baud Rate"** should match each other. For a new Xbee module, follow this procedure to query.
8) Select com port in section : "PC Settings"
9) Baud Rate set to 9600 (only for the new Xbee module , set the value to your case)
10) Flow Control : NONE
11) Data Bits : 8
12) Parity : NONE
13) Stop Bits : 1
14) Enable API : Uncheck
15) Click "Test / Query"
If we had set the Baud Rate to other value, then we should change to it.

When everything comes to the right place, this window will appear. If we see any other kind of message window, it means something wrong even we see an OK on it. It is not ok without this message and window. With the same setting, if we only got this occasionally , it means the communication is unstable. We may need to use an XBee adapter with a better quality , or try another XBee. The modem type and firmware version means the firmware be programmed in this XBee. It is possible to have other types of firmware , depends on your usage for XBee. If something goes wrong , then we will get this window. For most cases , it can be solved by just changing the Baud Rate and un/check API Mode. We will have a detailed information about this latter.
The wrong firmware in XBee will also result in this kind of message. If the setting and firmware are correct , then the hardware may have problem. In this case , check the adapter first then XBee. A hand made , bad quality Xbee adapter may result in this. Although the specification can not tell this , but XBee Pro apparently has a better tolerance on power source and adapter than normal XBee.The next section will discuss the four main tabs in X-CTU and information about using XBee with Arduino.

X-CTU User Interface:
Connecting & Configuring XBee for the project:
First, insert the xbee module in to the Xbee USB adapter. Next, we'll need to figure out which serial port (COM) we are using. Plug in the FTDI cable, USB adapter, Arduino, etc. Under windows, check the device manager, look for "USB Serial Port" Next we'll need to open up a terminal program. Windows comes with Hyperterminal, so just use that. Its under **Start->Programs->Accessories->Communications->HyperTerminal**. If we are running a different operating system just use whatever terminal program is available for it, such as ZTerm, minicom, etc. When we open it up, it should ask us for a new connection. Lets name it "xbee" Next We will select the COM port from the drop down menu, in my case its COM4. Next, set the properties. We select **9600 bps, 8 bit, No parity, 1 stop bit and no flow control.** Some programs may call this (9600 8N1). If the XBee has been configured for a different baud rate, of course, we should use that. We will get a blank screen that says "Connected" in the bottom left corner. Now, change the setup by selecting **File->Properties** and then going to the **Settings** tab and clicking the **ASCII Setup** button. Make sure we are sending line ends with line feeds and also echoing local characters Now type in **+++** (three plus signs) in quick succession. If the XBee is connected up properly we will get an **OK** in response If we got an OK that means the XBee is powered and wired up correctly! If its not working, check:
1. Try again, be sure to wait 10 seconds between each attempt at typing in +++ and type the +'s quickly
2. Is the module powered? Green LED should be blinking
3. Are RX & TX swapped?
4. Do we have the correct baud rate? By default it should be 9600 baud 8N1 no hardware handshake but if it has been used for something else the baud rate might be different . Next try typing in +++ (receive **OK**) and then **AT** and press return to get another **OK** This is basically how we can configure the XBee, by sending it AT commands (they all start with AT for ATtention). After a while, the XBee times out of configuration mode and goes back to passthrough connection mode. So if we want to get back to config mode, just type in +++ and it will start responding again.

Configuring Xbee for the project:

In Robot, we need to fix the Xbee with the destination address as the Base station xbee. Similarly in all other Swarm Robots all Xbee will contain the base station Xbee's Serial number as the Destination address. Base station Xbee can be configured to transmit to anyAddress as we are going to use this communication as the one way i.e swarm robots to base station.