M.A.G.U

Mobile Autonomous Grasping Unit

By

Richard Nigaglioni

4/23/1998

Table of Content

Abstract

Have you ever been in an office and needed to take some documents to someone else but you are too busy to get up to take them? Well MAGU could be the first step to designing a robot that can grasp the document and take it to that other person or place. I know it has been done before but perhaps not under $400. MAGU is an autonomous robot that moves with the use of tracks and grabs thing with its hand. Lets hope the construction of MAGU is a fun journey.

Executive Summary

The robot you are about to learn about has been designed to move small objects from one place to another and to avoid any obstacles it encounters. The robot's platform acts much like a tank and can move forward, backward and in 360 degrees of rotation. Two RC motors and proper gearing are used to move the platform. The arm of the robot allows the robot's hand to be moved up and down while keeping the hand in a horizontal position at all times. The hand is made out of wood and is controlled by a servo, which allows the hand to be closed with good accuracy. The robot is loaded with IR sensors that allow the robot to get an idea of its surroundings and weather an object is there to grab. The robot is controlled with the Motorola MC68HC11 board and the ME11 expansion. The programming of this board it's done with the reduced version of the C programming language, IC.

Introduction

MAGU is a robot that will help people by moving objects from one place to another. MAGU has a base that moves forward, backward, and in 360 degrees of rotation about its own axis. The base moves using treads from a TONKA toy, similar to a tank. The Robot has an arm and a grabbing hand used to grab the object the robot is going to move. IR sensors as well as bumper sensors are used to allow the robot to have an idea of its surroundings so that it can properly detect the object it's going to grab and avoid large obstacles. If time allows, the robot will carry a microphone that will be used to detect the presence of loud noises that alerts the robot that there are humans present and to be careful when it moves. The robot's main controlling unit is the M68HC11 with the ME11 attached. The knowledge I had of this board made programming it easy. The programming language used was IC. IC is a scaled down version C that was adapted to work with boards like the M68HC11. Now the fun part of describing the details of the robot begins!

Mobile Platform

The MAGU platform was designed to be similar to a tank. The wheels and Treads were taken from a TONKA toy that I bought at Toy's R Us for $17.00. The platform was built using wood and the t-tech-milling machine. I picked the tank idea because it gives the robot a faster turn on its own axis of rotation. This allows the robot to turn without moving forward or backward. The platform is 12.5 inches long by 9.5 inches wide by 4 inches tall. The size of this platform was chosen so that I would have enough internal space to place the motors, motor drivers, batteries, and extra weights for the balancing of the robot and the arm. Figure 1 shows the basic platform design. Pieces of wood were cut and used to form the IR and LED mounting space (see Figure 1). The EVBU board and arm will be mounted on top of the platform along with the on/off switch, IC loading selector, and a microphone device (note: the platform is designed to hold more LED and IR sensors if wall following is desire in future projects.)

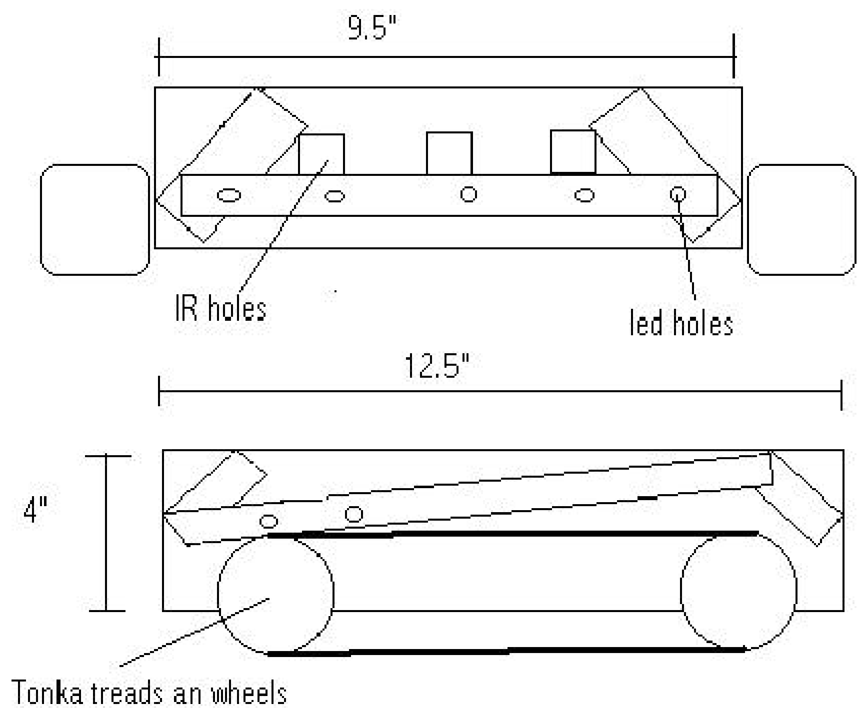

Figure 1 (Platform)

Arm and Hand Design

I got the arm from one of those self-adjusting arm lamps. The arm has two springs that force the arm to be stretched and to keep the hand that is placed at the end in a horizontal position at all times. The arm is placed at the front end of the robot and it's lifted and lowered by a motor/gear combination taken from the back wheels of an old RC car. The motor/gear combination is placed at the back end of the robot and a wire, attached to the end of the arm, is taken from front to back under the robot and winded on the motor/gear combination to ether lift or lower the arm (see Figure 2). This motor/gear combination is controlled by a motor driver described later on this report.

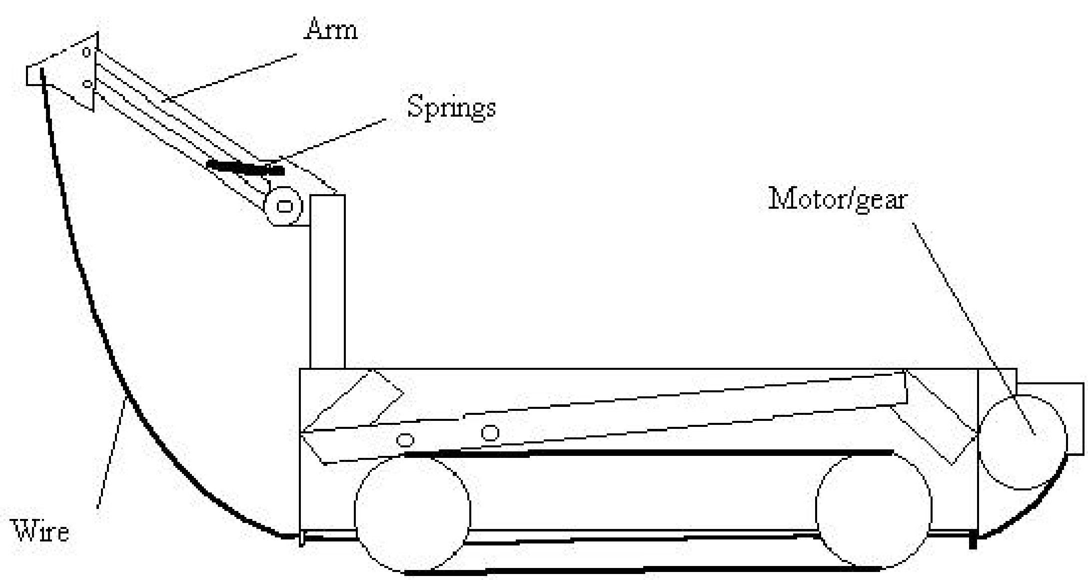

Figure 2 (Arm and Motor Placement)

The hand, made out of wood, consist of two pieces. Where one is stationary, holding an IR sensor, and the other, with the LED, is moved by a servo to open or close it (see Figure 3). The servo is mounted on the top of the hand and is controlled by pin A3 of the EVBU board. The servo it's powered by a 7.2V battery. The final additions to the hand were two sections of rubber that were placed inside the two pieces of grasping wood to stop objects from slipping.

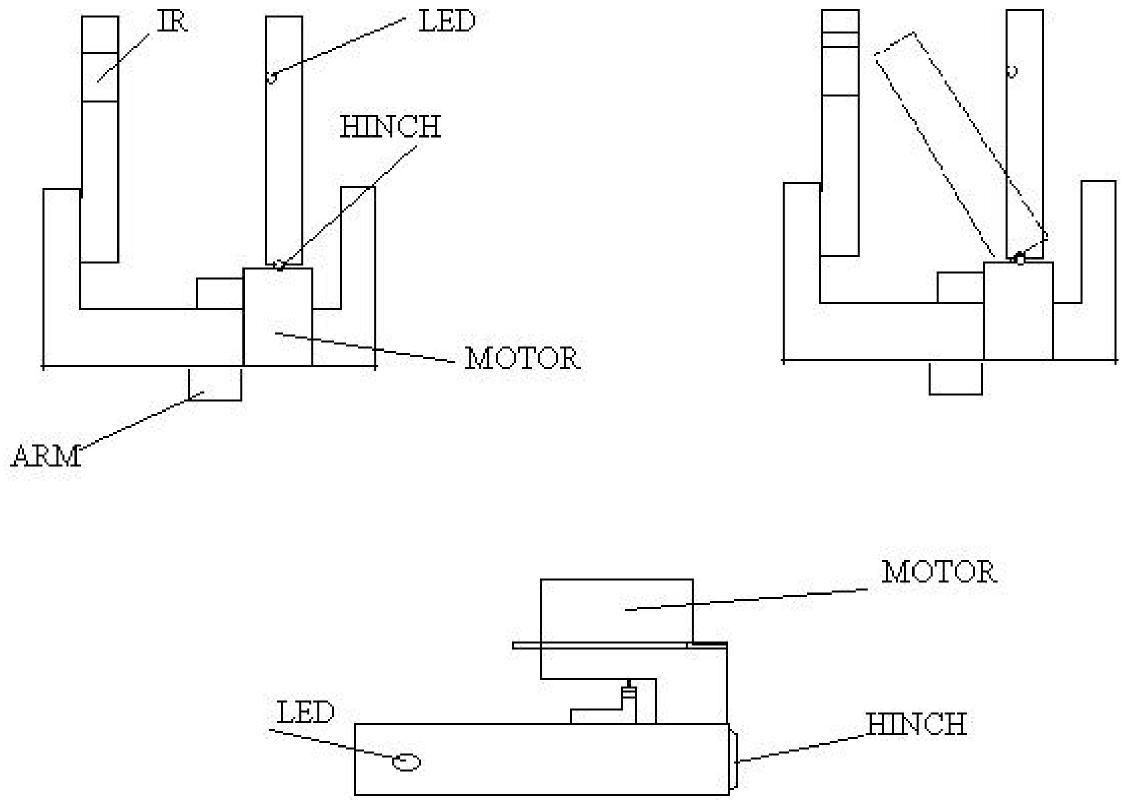

Figure 3 (Hand and Servo)

Platform Wheel Actuation

The platform wheels are moved by two 7.2V power motors that I bought from the Hobby Store at Gainesville. The motors are controlled by motor drivers (see Figure 4 for circuitry) allowing the motors to draw higher currents. The motor driver's TIP120 are cooled by a heat sink to prevent fires. Pins A5 and A4 from the EVBU board are used to controlled the motor's directions. The motor's speeds are controlled by pin D5 and pin D4 using a rectangular pulse wave. The gears that I used were purchased at the Hobby Store and they were mounted in a wheel rod from the inside of the platform (see Figure 5).

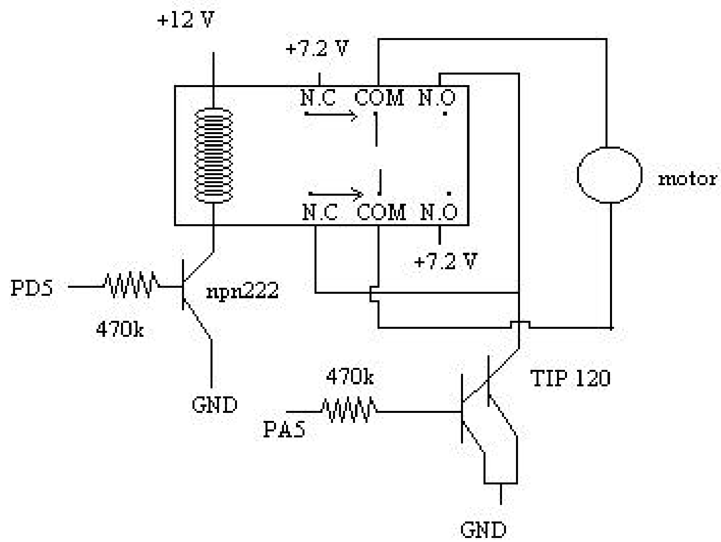

Figure 4 (Motor Driver)

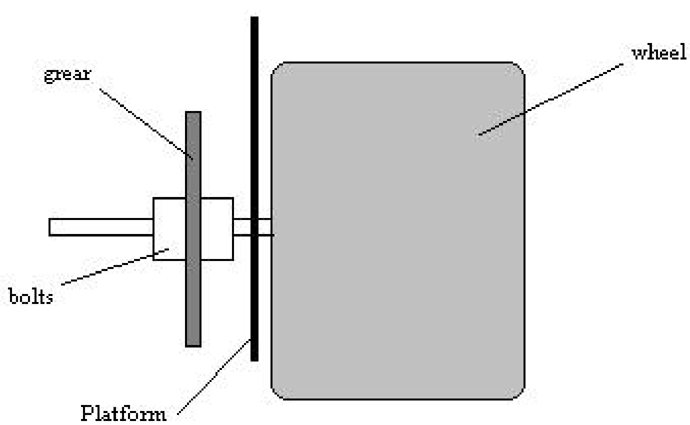

Figure 5 (Gear Mounting)

Sensors

The robot has two primary sensor types: IR, and bumper. The IR sensors along with a set of IR emitters are in the front of the robot to detect if there is an object in front of the robot. If an object is detected, then the robot will avoid the object. In addition to the front IR sensors, the robot will have one IR sensor mounted on the hand to detect if an object is on a position to be grabbed. The IR sensor will be connected to the analog port on the EVBU board, and the leds will be connected to an output port (0x7000) build into the ME11. For the case where the IR sensors fail to detect a wall, two bumper sensor are placed on front of the hand since the hand is most likely to encounter an object before the platform does. These bumper sensors will be connected to the analog port as well. Our final sensor is the microphone that will be use to detect the presence of noises. This microphone sensor was design by Danniel Copeland and it basically amplifies and extends the microphone input so that the EVBU can read it (see Figure 6 for circuit design). The robot will be programmed to a different behavior if it detects loud noises. The microphone used was purchase from Radio Shack. The microphone sensor will also be connected to the analog port. If you are wondering where I got that many analog inputs, well the analog ports PE0, and PE1 from the EVBU were expanded using an analog multiplexer to get a total of 16 analog inputs. The MUXs will be controlled by another output port from the EVBU board. This idea was taken from the TALRIK manual (page 13, Figure 5 MRSX01 circuit diagram). Figure 7 has the placement of the sensors.

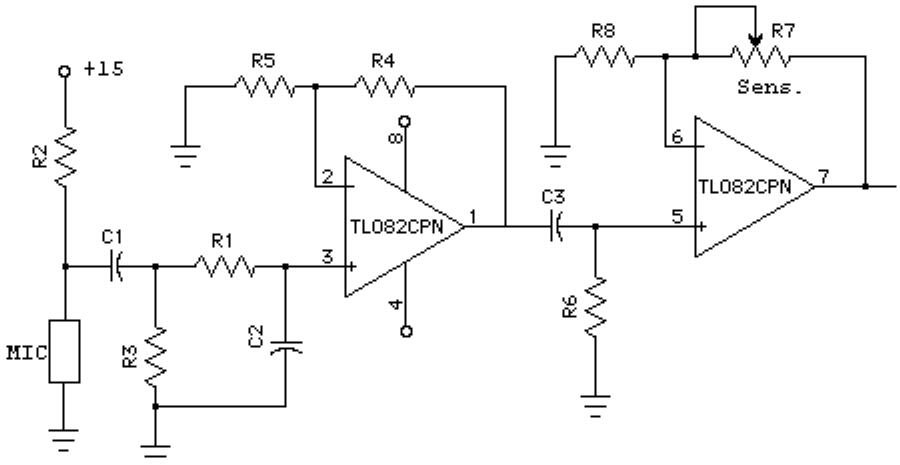

Figure 6 (Microphone amplification circuit)

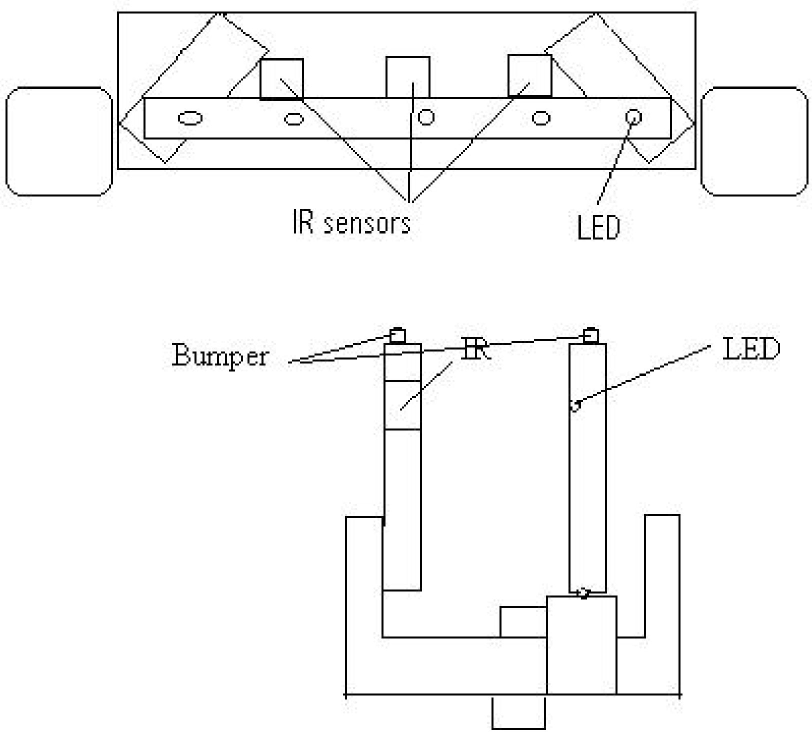

Figure 7 (Sensor Placing)

CPU board

The brain of the robot is the M68HC11 from Motorola. This chip and its board were chosen along with its expansion, the ME11, because of its easy configuration and because I was familiar with it working from previous classes. The working of this chip will take a whole paper to explain, but to get an idea, this board with the ME11 attached contains input, output, and analog ports, as well as two motor drivers, and 32k of RAM. The program that was used to program the robot was IC.

Power

A 12V battery powers the board and sensors, while a 7.2V battery powers the motors. The robot has a simple ON/OFF switch and a switch to change between downloading mode and IC execute mode.

Behaviors

Due to lack of time, the robot had only two behaviors, obstacle avoidance and grasping if an object it's detected by the hand. Code for these two behaviors is provided at the end of this report.

Experimental Data

The IR sensors used by the robot were ranged from 88 to 130 as read from the analog port. The servo at the hand had an angular placement between 35 to 95 degrees. And finally, the motor's speed ranged linearly from 100 to 40 percent where it declined exponentially.

Conclusion

Although the Robot was not fully finished, I am pleased to say that the two main functions that I wanted to accomplish were completed. These two functions were collision avoidance and grasping. During the process of designing and building the robot, I encountered several problems that delayed the robot's construction. The main problems I encountered were how to keep the robot's motor drivers from getting too hot, how to manage the voltage distribution, and how to communicate with my board since Windows crashed or it didn't detected the EVBU board. Solving these obstacles took more of my time that I expected but at the end with the help of the TAs, especially Scott, they were resolved. For the motor, a heat sink was used, for the voltage distribution, a breadboard was used, and for the communication, the computer needed to be in safe mode dos prompt. This robot was a fun project and one I will never forget.

Note: Special thanks to the TAs and Dr. Arroyo for their support and help and also to my friends Lars, Tom, and Jose for their ideas and support!

Appendix

```
/*Richard Nigaglioni*/
/*sensors*/
int FLeft_Sensor, FRight_Sensor;
int Hand_object;
/*Motors*/
/*ports*/
int Led_Port=0x7000, Sensor_Port=0x5000, Motor_Port=0x4000;
/*Variables*/
/*functions*/
void main()
{
servo_on();
start_process(IR_LED_ON());
start_process(SENSORS_ON_LOOP());
start_process(Avoid());
start_process(Close_Hand());

}
void Turn_Right()
{
motor(01,-50.00);
motor(00,50.00);
wait(500);
}

void Turn_Left()
{
motor(01,50.00);
motor(00,-50.00);
wait(500);
}

void Foward()
{
motor(01,70.00);
motor(00,70.00);
}

void stopall()
{
motor(01,00.00);
motor(00,00.00);
```

```
wait(1000);
}

void Backward()
{
motor(01,-70.00);
motor(00,-70.00);
}

/*obstacle avoid*/

void Avoid()
{
while(1)        {
        if   (FLeft_Sensor<=130   &&   FLeft_Sensor>=110   &&   FRight_Sensor<=130   &&
FRight_Sensor>=110) {
                Backward();
                wait(500);
                stopall();
                Turn_Right();
                Turn_Right();
                Turn_Right();
                stopall();
                Foward();}

        else if (FLeft_Sensor<=130 && FLeft_Sensor>=110)     {
                Backward();
                wait(500);
                stopall();
                Turn_Right();
                stopall();
                Foward();}
        else if (FRight_Sensor<=130 && FRight_Sensor>=110)   {
                Backward();
                wait(500);
                stopall();
                Turn_Left();
                stopall();
                Foward();}
        else
                Foward();
        }
}

void Close()
{
```

```
servo_deg2(95.00);
}

void Close_Hand()
{
while(1)        {
        if (Hand_object<110)    {
                close();
                stopall();
                wait(1000);
                Backward();
                foward();
                }
        }
}


void IR_LED_ON()
{
while(1)
        poke(Led_Port, 0xff);
)



void SENSORS_ON_LOOP()
{
/*reading of sensors*/
while(1)        {
        poke(Sensor_Port,0x0);
        FLeft_Sensor = analog(0);
        poke(Sensor_Port,0x4);
        FRight_Sensor = analog(0);
        poke(Sensor_Port,0x6);
        Hand_object = analog(0);
        wait(100);
        }
}

void wait(int milli_seconds)
{
long timer_a;
timer_a = mseconds() + (long) milli_seconds;
while (timer_a > mseconds()) {
        defer();
}
```

```
}
```