The Miracle

Mike Mattress
EEL 5666
Dr.Arroyo
08-02-98

Table of Contents

Abstract

There are many issues one has to address in the designing and implementation of a autonomous mobile robot that climbs stairs. Some of these issues include deciding upon a method of execution, design of a suitable platform and method of maintaining balance and position once upon a stair. The Miracle turned out to be a strong attempt at achieving its goal of climbing stairs. This final report will step one though all the decisions and steps to it's design and construction.

Executive Summary

The Miracles soul function is to climb stairs. It mimics how people walk stairs, one foot at a time. People but all their weight on one foot while lifting the other onto the next stair. Once both feet are firmly planted the person shifts their weight from back to front. Now the back foot is free to be lifted to the next stair.

The Miracle, although without feet, uses this same idea. It is constructed with two main body parts. The first is positioned in the front half of its body and the second is to its rear. To climb stairs it will first lift the front half of its body onto the next stair while its weight is on the rear half of the body. Once the front is upon the next stair the weight will be shifted to the front half. This will allow the rear to be lifted to the same stair as the front. The Miracle can then complete its task by simply driving forwards until fully on the taken stair and begin the process again. In addition to the two body parts previously mention The Miracle will make use of a movable weight, motors, wheels, push switches, gears, chains, and a Motorola 68HC11 EVBU board with a Mekatronic ME11 Expansion board for its intelligence.

Introduction

This final report will step by step walk one through the creation of The Miracle, the first attempt at a stair climbing autonomous mobile robot. Unlike most robot designs, The Miracle addresses many new issues to robot design. Size requirements, motor strength, gear ratios, weight limitations, and self-awareness are a few of these considerations. Each issue addressed by this robot will be discussed at length so that if another attempt were to be made at climbing stairs one could learn from the design of The Miracle.

Integrated System

To begin with any robot that is to be autonomous will need some form of artificial intelligence. For this project the Motorola 68HC11 board with the Mekatronic ME11 Expansion board were chosen. This combination of electronics was chosen primarily due to use friendliness. If one has taken EEL4744 they are probably the most familiar and comfortable with the 68HC11 EVBU. The ME11 is a pre-designed board that only requires basic soldering to complete. It expands the 68HC11 with 32K of memory, 2-1A variable speed and direction motor drivers, two input/output control lines and an 8-bit input/output port, which was not used.

Additionally, upon the free half of the 68HC11 board 6 relays, 1-8 bit input port, and 1-8 bit output port was added. Both the input and output ports are clocked by the 1 of the two control line pairs of the ME11. The input port was wired to 8 push switches and the output port is used to control the 6 relays. The relays are used to de-multiplex the two motor drivers upon the ME11.

It was decided that de-multiplex the 2 motor drives would of benefit. With this hardware implementation 4 additional motor drivers would not be needed for the 6 motors. Also, when the program was coded the motors could be turned on and never turned off. Only the speed and direction would have to be changed. To select motors, one will only need to write to the output port controlling the relays. One must however be careful how many motors are on at one time. The motor driver chip can only source 1 am per driver so running all 6 motors simultaneously could cause a failure or even damage of the hardware. However, climbing stairs only required two motors to be on at a time.

To supply the needed power 14 AA batteries will be used to supply. Reset and on/off will be wired to switches installed on the robot's surface for easy access. All connections to the 68HC11/ME11 will be plug in style. Communication to the electronics will be achieved with a serial wire from a computer. The software is written in assembly language, compiled with AS11, simulated with SIM11 and downloaded using HyperTerminal. The program will be tested on board in RAM but if small enough may finally be loaded to EEPROM.

Mobile Platform

The mobile platform for this particular robot had to be very well engineered. Half of all time spent on this robot went into its design and construction. As explained before the body will be made two halves; a front and a back with a weight shifter on top. These three pieces will be made of wood that has been milled by the T-tech machine in the IMDL lab. The design for the milling was draw in AutoCAD14.

The front piece has to be made as small as possible. First, its height has to be small since the rear has to lift it a distance equal to the height of the stair plus its own height. The average stair is less than 10 inches wide so the robot itself must be less that 10 inches long so as to fit entirely upon a stair. The average stair has a rise that is 9 inches or less also, so the robot must be able to lift the front by its height plus 9 inches. If the height of the front is kept within 3 inches the robot will stand 12 inches tall. The rear and front together must be less than 10 inches and each will therefore be less that 5 inches. This means that the robot must be able to maintain balance of more than half its weight 12 inches high upon a wheelbase of 4 inches and be able to drive forward in this position.

The front must also wrap around the rear to provide for a contact between the two. This contact will allow the rear move with respect to the front and vice versa. On the front face of the front half of the body there is a hole for access to the serial connector of the 68HC11. Also, there are two push switches used to detect the rise of the stair in front of the robot and allow for the robot to square up against the stair. The width is kept to a minimum for no other reason than to keep the weight to a minimum. Upon the front half there is a track that the shift weight will use to move the weight forward and backward to change the center of gravity. Also, on the top are mounted the reset and on/off switches. Inside the electronics are mounted upside down and on the side are mounted two motors with wheels.

The rear body part must be 12 inches tall to achieve the needed lift. Its base is 4 x 4 inches. On its front face are two push switches that are used to detect when a stair is under the front and also to allow the robot to square again against the stair. Inside are mounted two motors and wheel and an additional motor geared to an axle that will turn the chains to lift the front. This will be explained further later on.

The weight shift is small bucket that contains the batteries and bee-bee shots for additional weight. It also has a motor attached to a gear that will drive upon the track of the front body part.

The entire body has been designed so that the center of gravity will fall close to the midpoint. This is important so that the weight can be shifted easily from front to back and vice versa. Clearance between the electronics and lift motor gearing had to be allowed for. The gearing is covered so that wires will not fall into the teeth when in the down position. Holes one the sides of the rear have been made to prevent friction between the original Lego chain and the wood and also so that push switches can be installed upon the connection between front and back to detect maximum and minimum elevation. Also there are mounting for two push switches for maximum and minimum weight shifting. Finally the front is made to come down as close to level with the rear as possible. This is important because at one point the robot will he half on the stair and have to drive forward to get completely on the stair. If the weight difference between front and rear wheels is too large the robot will surely get stuck. The height difference for The Miracle is 3/8 inch.

Actuation

Other than the lift motor the small servos sold by Mekatronic will be used for all actuation. The servos have been hacked so that they will behave as motors. They are used for the 4 wheels and the weight shifter. The lift motor however has to have more strength than the servos can provide. The Miracle made use of a timing motor obtained by a fellow student. The motor has to be capable of lifting 4 pounds at 18V and draw at most 1A stalled otherwise electronic hardware changes must be made.

The lift motor is geared 1 to 1 to an axle that runs to the sides of the rear. There the axle is attached to rubber wheels were a pull chain wraps around it. The pull chain makes a single twisted loop so that lift is obtained on bot far extremities of its path. The rubber wheel prevents slipping of the chain when torque is applied. Originally, Lego gears and chains where to be use however it was found the gears slipped and stripped and the chain disconnected under the weight being lifted. The Lego wheels did however work nicely due to their rubber wheel for friction and small size.

The final lift actuation although unusual worked nicely. It was tested to lift over 5 pound without stalling or breaking. The only difficulty was that the chain did slowly stretch and thus generate play. This play causes the front loose its level. Another less significant problem was due to the front hanging by chains. When driving the front tended to swing slightly. This is a possible problem because of the robots balance is delicate. It seemed possible for the robot to actually knock itself over although it has never experienced

Sensors

The Miracle made use of the simplest of sensors, push switches. No analog sensors were used or needed. Some of the push switches where however used in a very unique way. Four were merely used to detect when the robot was against the stair. These switches were of the type having a small arm with a roller wheel at the end. This roller ball allowed the switch to be dragged up the face of a stair while maintaining contact. The robot would be able to tell when above the stair when the sensor released. Also, with these sensors since in pairs the robot can tell how it positioned relative to the stair it has contact with. If one sensor is active while the other is not the robot can tell which side of itself has contact and turn accordingly to square with the stair. It will know when square when both switches are active and it will also know when contact is lost when both switches are not active.

The remaining 4 push switches are identical except that they lack the roller wheel at the end. They are use for self-awareness. The robot must never try to exceed its maximum and minimum height and weight shift. If it did the robot would actually break itself. In addition to self-preservation the robot uses these sensors to monitor steps in the climbing of stairs. For example, Once the front has a stair underneath it the weight is shifted forwards and stopped when at the front maximum location. This sensor thus signals the 68HC11 that the weight shift is completed and the next phase may now commence. Another example occurs when the robot is lifting the front. The 68HC11 will poll the front push switches as well as the maximum height switch. This is necessary in case the stair is taller that the robots maximum height. If the maximum height switch is activated before the front switches are deactivated the 68HC11 then will realize the stair is too tall for it and quit.

Behaviors

Upon reset the robot must initialize itself. This is the robots first behavior. It consists of moving the weight shift all the way back and lowering. Upon completion the robot now can move freely. It must now find a stair and to do so it simple drives forwards until it hits something. Assuming it is a stair the robot attempts the square up against the stair. This is a form of obstacle seeking. Once against the rise of a stair the robot begins its sequence to climb the stair, which is as follows;

1. Lift front until stair is cleared or until maximum lift is reached.
2. Drive forward until stair rise hits front face of rear.
3. Shift weight from rear to front.
4. Lift rear fully (inverse action to lifting front).
5. Drive forward until front face hits next stair rise.
6. Shift weight from front to rear
7. Repeat steps 1-6.

Since the platform design was very elaborate and time consuming other behavior implementation were not possible. However, ideas for stair/wall differentiation using IR emitters/detectors were considered. The IR system would have to be placed on the robot's front at a height great enough so that the IR would see weather or not a wall was present. If it did not see a wall the robot could assume a stair.

Experimental Layout and Results

Unfortunately, a problem arose with the electronics at the end of the semester. The voltage regulator caught fire and several traces burned out of the ME11. The exact cause has yet to be determined however two ideas exist for the cause. The first possible cause is that the voltage regulator overheated and shorted internally causing the surge. The second possibility is that the relays when switched off generate a back EMF that damaged the voltage regulator. To correct these problems a heat sink should be used for the voltage regulator and a diode resistor series should be put across the relays switch to diffuse the back EMF.

Before the fire the robot showed much promise of success. Each sequence to climb a stair had been tested individually and was being assembled into one. The robot was able to lift the needed weight and balance it at all times. The lift weight however turned out to be heavier than expected. Since the motor had to lift a lot of weight it had to be strong and thus heavy itself. When the robot was upon a stair resting on its front wheels this weigh with the rear wheels and there motors generated a back torque of approximately 2 pounds at 3 inches. To overcome this the robot needed an opposing torque generated by the shift weight. The shift weights moment arm however was about 2.5 inches so the weight had to be in excess of 2.4 pounds to assure that the robot did not fall over backwards when at this point.

Conclusion

Although The Miracle did not live up to its name and never climbed a stair it came very close. And in doing so is sure to serve as a guide for the next to attempt the same feat. All of the challenges of climbing stairs were resolved by this robot. It still has the potential to climb stairs. However, lack of funds and remaining time has ceased its work. Much was learned in its design. AutoCAD14 and the T-tech machine have become very familiar tools. Hands on electronic hardware design and implementation were successfully achieved. Problem solving on many levels and the challenge of being inventive as well as resourceful were also experienced.

Finally, the results of not being fully aware of the specifications of the electronics used were learned. This alone, although disappointing at the time, is what I shall take from this class above all else.

Appendices

```
BIT1    EQU     %00000010
BIT0    EQU     %00000001
BIT4    EQU     %00010000
BIT6    EQU     %01000000
INV5    EQU     %11011111
INV6    EQU     $10111111

STACK   EQU     $0041
BASE    EQU     $1000
TOC1    EQU     $1016
TOC2    EQU     $1018
TOC3    EQU     $101A
TOC4    EQU     $101C
TCTL1   EQU     $1020
TCTL2   EQU     $1021
TMSK1   EQU     $1022
TFLG1   EQU     $1023
TFLG2   EQU     $1025
TMSK2   EQU     $1024
BAUD    EQU     $102B
SCCR1   EQU     $102C
SCCR2   EQU     $102D
SCDR    EQU     $102F
SCSR    EQU     $102E
TCNT    EQU     $100E
PACTL   EQU     $1026
PORTD   EQU     $1008
DDRD    EQU     $1009

SENSOR  EQU     $6000

LIGHTS  EQU     $7000
ON      EQU     $10101010
```

```
OFF         EQU       $00000000

MOT_S       EQU       $6000
FFD         EQU       %11110000
REAR_M      EQU       %11111100
RL_M        EQU       %11111110
RR_M        EQU       %11111101
FRONT_M               EQU       %11110011
FL_M        EQU       %11111011
FR_M        EQU       %11110111
LIFT_M      EQU       %11101111
SHIFT_M     EQU       $11011111
**************************************************
* GLOBAL VARIABLES
**************************************************
            ORG       $C000
ON_M        RMB       2         ; ON TIME OF MOTOR
OFF_M       RMB       2         ; OFF TIME OF MOTOR
*                               ; ON+OFF=$9800 FOR 51.4 Hz
SWITCH2     RMB       1         ; USE TO SWITCH BETWEEN ON AND OFF
SWITCH3     RMB       1
**************************************************
* LOCATOR TABLE
*                     SPEED               DIRECTION
* MOTOR1              PA6(OC2) PD4
* MOTOR2              PA5(OC3) PD5
**************************************************

*-------------------------------------------------
* INTERRUPT VECTORS
*-------------------------------------------------

    ORG    $00DC
    JMP    MOTOR1

    ORG    $00D9
    JMP    MOTOR2

*     ORG    $FFE4              ;SIM11
*     FDB    OC3_ISR

*     ORG    $FFE6              ;SIM11
*     FDB    OC2_ISR

**************************************************
* MAIN PROGRAM
**************************************************
    ORG    $8000
            LDS       #STACK
            LDX       #BASE
*-------------------------------------------------
* INITIALIZE
*-------------------------------------------------
* TURN OFF ALL MOTORS
            LDAA      #$FF
            STAA      MOT_S

* SET PORTD TO OUTPUTS
            LDAA      #$FF
            STAA      DDRD

* OC2 AND OC3 OUTPUT CLEARED (MOTOR DRIVERS SPEED)
    LDAA   #%10100000
            STAA      TCTL1

* SET MOTOR DIRECTIONS (?BIT5=LEFT ?BIT4=RIGHT)
            LDAA      #%00      ;00110000
            STAA      PORTD

* SET ON AND OFF TIMES AND SWITCH
```

```
        LDD     #$8C00          ; SET ON TIME
        STD     ON_M
        LDD     #$1000          ; SET OFF TIME
        STD     OFF_M
        LDAA    #$01            ; INITIALIZE SWITCH TO 1
        STAA    SWITCH2
        STAA    SWITCH3

*--------------------------------------------------
* INTERRUPT ENABLES SETTINGS
*--------------------------------------------------
* ENABLE INTERRUPTS
    LDAA  #%01100000    ;ENABLES OC2 AND OC3 INTERRUPT (MOTOR DRIVERS)
        STAA    TMSK1

    CLI                     ;ENABLE INTERRUPTS
****************************************************
* MAIN BODY
****************************************************
*--------------------------------------------------
* DRIVE FORWARD
*--------------------------------------------------
        LDAA    #$00
*AGAIN  LDAA    #REAR_M         ; TURN ON REAR MOTORS ONLY
        STAA    MOT_S
HERE   BRA   HERE

* CONTROL SPEED AND DIRECTION
POLL    LDAA    SENSOR          ; POLL FRONT FOR HIT
        ORAA    %00000011
        BEQ     POLL
        LDAA    SENSOR          ; CHECK FRONT LEFT
        ORAA    %00000001
        BEQ     RIGHT
        LDAA    #FR_M           ; HIT ON LEFT FRONT TURN ON RIGHT REAR MOTOR ONLY
        STAA    MOT_S
POLL2   LDAA    SENSOR
        ORAA    %00000010       ; CHECK FRONT RIGHT
        BEQ     POLL2
        JMP     LIFT
RIGHT   LDAA    #FL_M           ; HIT ON RIGHT FRONT TURN ON LEFT REAR MOTOR ONLY
        STAA    MOT_S
POLL3   LDAA    SENSOR
        ORAA    %00000001       ; CHECK FRONT LEFT
        BEQ     POLL3
        JMP     LIFT
*--------------------------------------------------
* LIFTING BODY FRONT
*--------------------------------------------------
                        LIFT    LD
```

Table of

```
A           #LIFT_M             ; LIFT FRONT
            STAA      MOT_S
*CONTROL SPEED AND DIRECTION
POLL4       LDAA      SENSOR
            ORAA      %00100011           ; CHECH FOR MAX, LEFT FRONT AND RIGHT FRONT
            BEQ       POLL4
*DELAY
*--------------------------------------------------
* ELEVATED DRIVE
*--------------------------------------------------
            LDAA      #REAR_M                       ; REAR WHEEL DRIVE FORWARDS
            STAA      MOT_S
*CONTROL SPEED AND DIRECTION
POLL5       LDAA      SENSOR
            ANDA      %00001100           ; CHECK REAR LEFT AND RIGHT FOR RELEASE
            BNE       POLL5
*DELAY
*--------------------------------------------------
* SHIFT WEIGHT FORWARD
*--------------------------------------------------
            LDAA      #SHIFT_M                      ; SHIFRT WEIGHT FORWARD
            STAA      MOT_S
*CONTROL SPEED AND DIRECTION
POLL6       LDAA      SENSOR              ; CHECK IF AT FRONT
            ORAA      %10000000
            BEQ       POLL6
*--------------------------------------------------
* LIFTING BODY REAR
*--------------------------------------------------
            LDAA      #LIFT_M             ; LIFT REAR
            STAA      MOT_S
*CONTROL SPEED AND DIRECTION
POLL7       LDAA      SENSOR              ; CHECK IF DONE
            ORAA      %00010000
            BEQ       POLL7
*--------------------------------------------------
* DRIVE ONTO STEP
*--------------------------------------------------
            LDAA      #FFD                ; 4-WHEEL DRIVE
            STAA      MOT_S
*CONTROL SPEED AND DIRECTION
* DELAY UNTIL ON STAIR
            JMP       AGAIN
****************************************************
* INTERRUPT SERVICE ROUTINES
****************************************************
*--------------------------------------------------
* MOTOR 1
*--------------------------------------------------
MOTOR1      LDX       #BASE
            BCLR      TFLG1-BASE,X INV6             ;CLEAR FLAG
            LDAA      SWITCH2                       ; IS SWITCH2 ON OR OFF
            BEQ       SET2                          ; IF OFF SET
CLEAR2      LDD       TOC2                          ;
            ADDD      OFF_M
            STD       TOC2                          ; ADD ON TIME TO TOC2
            BCLR      TCTL1-BASE,X BIT6             ; OUTPUT=0
            LDAA      #$00
            STAA      SWITCH2                       ; TURN SWITCH2 OFF
            BRA       RT_OC2
SET2        LDD       TOC2
            ADDD      ON_M
            STD       TOC2                          ; ADD OFF TIME TO TOC2
            BSET      TCTL1-BASE,X BIT6             ; OUTPUT=1
            LDAA      #$01                          ; TURN SWITCH2 ON
            STAA      SWITCH2
RT_OC2      RTI
```

```
*---------------------------------------------------
* MOTOR 2
*---------------------------------------------------
MOTOR2  LDX     #BASE
      BCLR    TFLG1-BASE,X INV5         ;CLEAR FLAG
        LDAA    SWITCH3                 ; IS SWITCH ON OR OFF
        BEQ     SET3                    ; IF OFF SET
CLEAR3  LDD     TOC3                    ;
        ADDD    OFF_M
        STD     TOC3                    ; ADD ON TIME TO TOC3
        BCLR    TCTL1-BASE,X BIT4       ; OUTPUT=0
        LDAA    #$00
        STAA    SWITCH3                 ; TURN SWITCH3 OFF
        BRA     RT_OC3
SET3    LDD     TOC3
        ADDD    ON_M
        STD     TOC3                    ; ADD OFF TIME TO TOC3
        BSET    TCTL1-BASE,X BIT4       ; OUTPUT=1
        LDAA    #$01                    ; TURN SWITCH3 ON
        STAA    SWITCH3
RT_OC3  RTI
****************
```