USING INFRARED TRANSISTORS TO CONTROL A MOTORIZED TRACKING SYSTEM

Mohammad Fotouhi , William Cavey

Electrical Engineering Technology
University of Maryland Eastern Shore
Princess Anne, MD 21853

ABSTRACT

The purpose of this project was to provide experience with infrared detection and feedback control of a dual axis system for tracking purposes. The objective was to design and construct a detector using four infrared transistors that were mounted on a dual axis platform. The detector provides quadrant oriented information that is processed using a BASIC STAMP II microcontroller. The microcontroller analyzes the information and tries to keep the detector centered about the infrared source. As the infrared source moves, the detector follows immediately.

Interfacing of a microcontroller, A/D converters, operational amplifiers, infrared transistors and stepper motors was required. Development of a software program to process data and determine tracking controls was necessary. The program was written in BASIC STAMP II software using a PC. Downloading the program into the microcontroller was accomplished through a serial interface. Real time feedback was available on the screen during the initial setup and testing of the circuitry. The paper discusses the project in detail.

INTRODUCTION

Design of this infrared tracking system was a project in our ETEE 499 undergraduate advanced design / research course. There were many individual components that students were already familiar with or could easily learn to use. These components were NPN transistors, infrared (IR) transistors, analog to digital converters, Basic Stamp II microcontroller, voltage regulators, operational amplifiers and stepper motors. This paper shows how these components work as a closed loop system. There is an input section, processor and software, output circuitry and a mechanical section.

INPUT CIRCUITRY

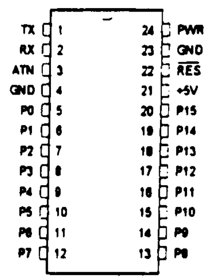
Input circuitry is used to detect infrared light (*Figure 1*). This is accomplished using four darlington infrared transistors (LF14f1) with a peak wavelength of about 850 nm. The collector is connected to the regulated five-volt source, the emitter goes to a 20K resistor to ground and the base is left unattached. When light strikes the lens of the transistor, current flows through the 20K resistor and provides a voltage drop across it proportionally to the light intensity. An MC3303 quad op-amp, which only requires a single supply, is also used. The outputs from the emitters of the transistors are connected to the positive inputs of each op-amp. The op-amps are configured to have unity gain and provide a buffer between the transistors output and the A/D converters. Four individual A/D converters receive voltage from the op-amps and output 8 bit data to the microcontroller. The ADC0831CCN is an 8 bit serial output A/D converter that was chosen because it is easily interfaced to our microcontroller. It is set up so a 0 to 5 volt input provides a 0 - 255-binary output. The Basic Stamp II is well suited for interfacing to the ADC0831CCN. The Stamp has four pins dedicated to handling the chip select of each converter. The Basic Stamp II uses only one line that is shared by all four A/D converters for data inputs to the Stamp. Also, only one clock line is shared by the A/D converters. This is possible because only one A/D is selected at a time; therefore, the data present on the data line is only from the selected A/D. This allows multiplexing of several A/D converters without dedicating more data and clock lines. This is important because on the Basic Stamp II there are only 16 I/O lines and they must be used wisely.

PROCESSOR AND SOFTWARE

The processor is known as the Basic Stamp II (*Figure 2*) which uses a 16C57 PIC as its central processing unit. There are 16 I/O lines, 2048 bytes of EEPROM and it performs 4000 instructions per second. The Basic Stamp II has output ports, OUTA through OUTD, which have predefined specifications. Each is assigned to a set of four pins, OUTA P0-P3, OUTB P4-P7, OUTC P8-P11 and OUTD P12-P15. The programming language is similar to Basic and, having only a few instructions, can easily be learned. After a program is written in its editor, it is downloaded through a serial port to the Stamp. A carrier board can be purchased for the Basic Stamp II which has a serial port and provisions for a 9-volt battery to supply power. The software written for this project has several sections that need to be explained, the input process, data conditioning, decision making and motor control outputs. The detail of the program is explained in the appendix. Variables and constants are assigned at the beginning of the program and all the A/D converters are disabled to make sure no data is on the data line. The input process starts with a single reading of all four A/D outputs. This data represents ambient values of light that are detected by the IR transistors and these values are assigned to variables. Poling of the IR transistors begins next. The values are read again from the A/D converters and then the ambient values are subtracted. These new values are also assigned to 8 bit variables. Conditioning of these values is done by assigning the 3 most significant bits of the last 8 bit variables to a new set of variables. This means the lower 5 bits is not considered (*Table 1*).

0	0	0	X	X	X	X	X
0	0	1	X	X	X	X	X
0	1	0	X	X	X	X	X
0	1	1	X	X	X	X	X
1	0	0	X	X	X	X	X
1	0	1	X	X	X	X	X
1	1	0	X	X	X	X	X
1	1	1	X	X	X	X	X

***Table 1*. Conditioning of Variables**



Pin	Name	Description	Comments
1	TX	Serial output	Connect to pin 2 of PC serial DB9 (RX) *
2	RX	Serial input	Connect to pin 3 of PC serial DB9 (TX) *
3	ATN	Active-high reset	Connect to pin 4 of PC serial DB9 (DTR) *
4	GND	Serial ground	Connect to pin 5 of PC serial DB9 (GND) *
5	P0	I/O pin 0	Each pin can source 20 ma and sink 25 ma.
6	P1	I/O pin 1	
7	P2	I/O pin 2	P0-P7 and P8-P15, as groups, can each
8	P3	I/O pin 3	source a total of 40 ma and sink 50 ma.
9	P4	I/O pin 4	
10	P5	I/O pin 5	
11	P6	I/O pin 6	
12	P7	I/O pin 7	
13	P8	I/O pin 8	
14	P9	I/O pin 9	
15	P10	I/O pin 10	
16	P11	I/O pin 11	
17	P12	I/O pin 12	
18	P13	I/O pin 13	
19	P14	I/O pin 14	
20	P15	I/O pin 15	
21	+5V **	+5V supply	5-volt input or regulated output.
22	$\overline{RES}$	Active-low reset	Pull low to reset; goes low during reset.
23	GND	System ground	
24	PWR **	Regulator input	Voltage regulator input; takes 5-15 volts.

* For automatic serial port selection by the BASIC Stamp II software, there must also be a connection from DSR (DB9 pin 6) to RTS (DB9 pin 7). This connection is made on the BASIC Stamp II carrier board. If you are not using the carrier board, then you must make this connection yourself, or use the command-line option to tell the software which serial port to use.

** During normal operation, the BASIC Stamp II takes about 7 mA. In various power-down modes, consumption can be reduced to about 50 µA.

FIGURE 2. Basic Stamp II

Using only these three bits, all decision making is made. This conditioning allows for 8 levels of light intensity instead of 256 levels. This reduction is necessary for this application to help eliminate mechanical oscillations. The mechanical movements do not have enough resolution to accommodate the IR transistors' and the A/D converters' response. The three bit variables are used to determine if any single transistor quadrant output of the detector head is greater than the rest, or if any two quadrants are equal and greater than all the rest or if they are equal (*Table 2*). There are nine possible tracking options that assign variables to control the rotation of motor1 and motor2. Actual motor controls are accomplished by looking up a data word from a list that corresponds to stepper motor control specifications and provides 3.75 degree half step. One word per motor is sent to two separate four-bit ports named OUTB and OUTC. OUTB and OUTC are chosen in this project strictly for convenience. Each four bit port corresponds to four control lines needed for each stepper motor. A variable is used to select and keep track of the last data word selected from the list. After each reading of the IR transistors, this variable is added to or subtracted from to make the motors rotate clockwise or counter clockwise.

OUTPUT CIRCUITRY

In order to observe the output response from the microcontroller, indicators are used. Pin 14 of the Basic Stamp II is connected to a LED through a 330 ohm resistor to ground which shows when the sensors are being poled. Pin 15 is also connected to a LED through a 330 ohm resistor to ground and shows when an infrared source is centered among the four IR transistors. This LED is called the 'lock on' LED.

Output to the motors is achieved through 20K resistors from pins of the Stamp that associate with OUTB and OUTC ports (*Figure 3*). These resistors are connected to the base of an NPN 2N2222A transistor which is attached to a second 2N2222A in a darlington configuration. The high gain of the darlington pair allows the transistors to act as on/off switches with very little drive current supplied by the Basic Stamp II. The emitters of the transistors are tied to the ground. Each collector of transistors goes to a separate coil of a stepper motor. There are four drive circuits and two 12 volt supply lines for each motor. When an output of the Stamp goes high (5 volts) it will turn on the drive transistors allowing current to flow from the 12 volt source, through motor coils to the ground. The order of turning on these drive transistors determines, the direction of the motor rotation, how fast it rotates and whether it half steps or full steps. This order is controlled by the Stamp and is covered in the motor control output section of the software. Another important design consideration of driving stepper motors is the suppression of voltage spikes created by the coils of the motors. There is a diode used across each of the four coils per each motor to eliminate the voltage spikes.

MECHANICAL DESIGN

The mechanical design (*Figure 4*) includes the following parts: stepper motors, detector head and support mechanisms. The stepper motors operate on 12 volts and have 7.5 degree full steps. Motor 1 is a larger motor used to rotate the detector head right and left. Motor 2 is a smaller stepper that moves the detector head up and down. The detector head is constructed of a small piece of PVC pipe 1" long and ¾" in diameter. A small plastic disk is made to fit into the PVC pipe. The disk has four holes, one in each quadrant of the disk approximately half way from the center to the edge of the disk. The infrared transistors are inserted into the holes and glued in place. The disk is placed inside the PVC so the lenses of the transistors clear one end of the pipe. A plastic infrared passing lens is added to the end of the PVC pipe covering the transistors. This lens helps to filter out visible light that the IR transistors may react to. The transistors are labeled 1 to 4 counter clockwise, facing the transistor lenses with 1 in the upper right-hand quadrant. Two small holes representing the horizontal axis are added to the PVC one between quadrants 1 and 4 and one between quadrants 2 and 3. They are located close to the lens end of the pipe. To support the detector head a 'U' shaped bracket is formed and corresponding holes added to align with the holes in the PVC. On the left side, facing the detector, the small stepper motor is attached to the 'U' support." The shaft passes through the support and into one hole of the pipe and is glued to the pipe. The other side of the 'U' support requires a small metal pin to act as a pivot. The pin is pushed through the 'U' support and into the second hole in the pipe. It can be glued on either end as long as it allows free movement of the detector head. The bottom of the 'U' support is attached to the shaft of the larger stepper motor.

OPERATION

As the circuit is powered, the Basic Stamp II begins to pole the four infrared transistors. An infrared LED is used to act as the source to be tracked. When the LED moves in front of the detector head, the detector head tracks the LED's movement. Tracking the LED works better in a dark room, because infrared emissions including room lighting can interfere with tracking the LED. As the detector head becomes centered, the 'lock on' LED will light indicating the source is centered between the IR transistors.

CONCLUSION

This project offered many educational opportunities. Several components were linked together in this closed loop system and it provided a great learning experience. Lessons learned include how to write a program in Basic Stamp II software and how to control a stepper motor with a microprocessor. Experience in software controlled feedback and data conditioning was also gained. Knowledge of infrared transistors and minimizing noise interference was also explored. This project provided challenges that promoted research and furthered interest in closed loop microprocessor controlled systems.

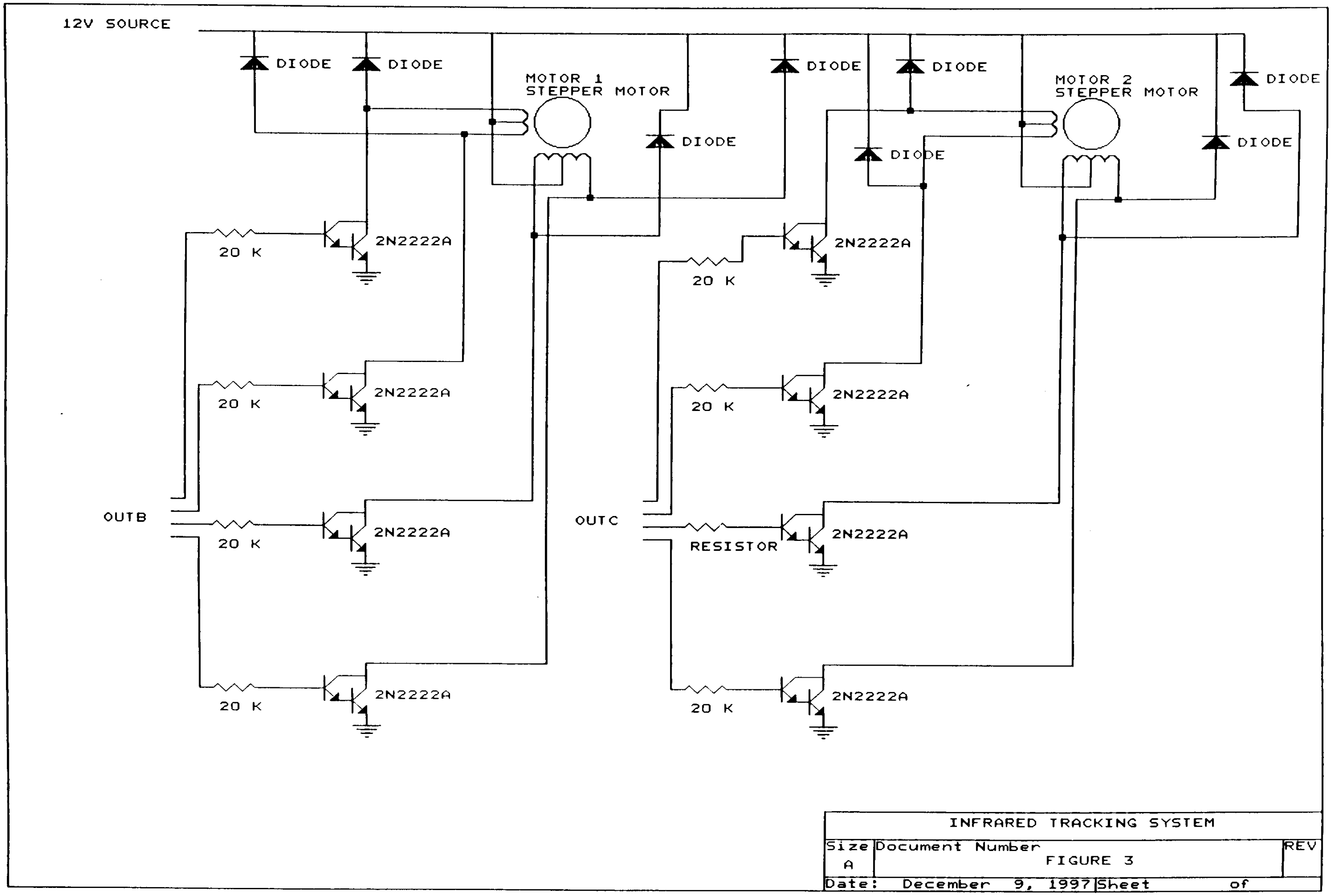


FIGURE 3. Driver for Stepper Motors.

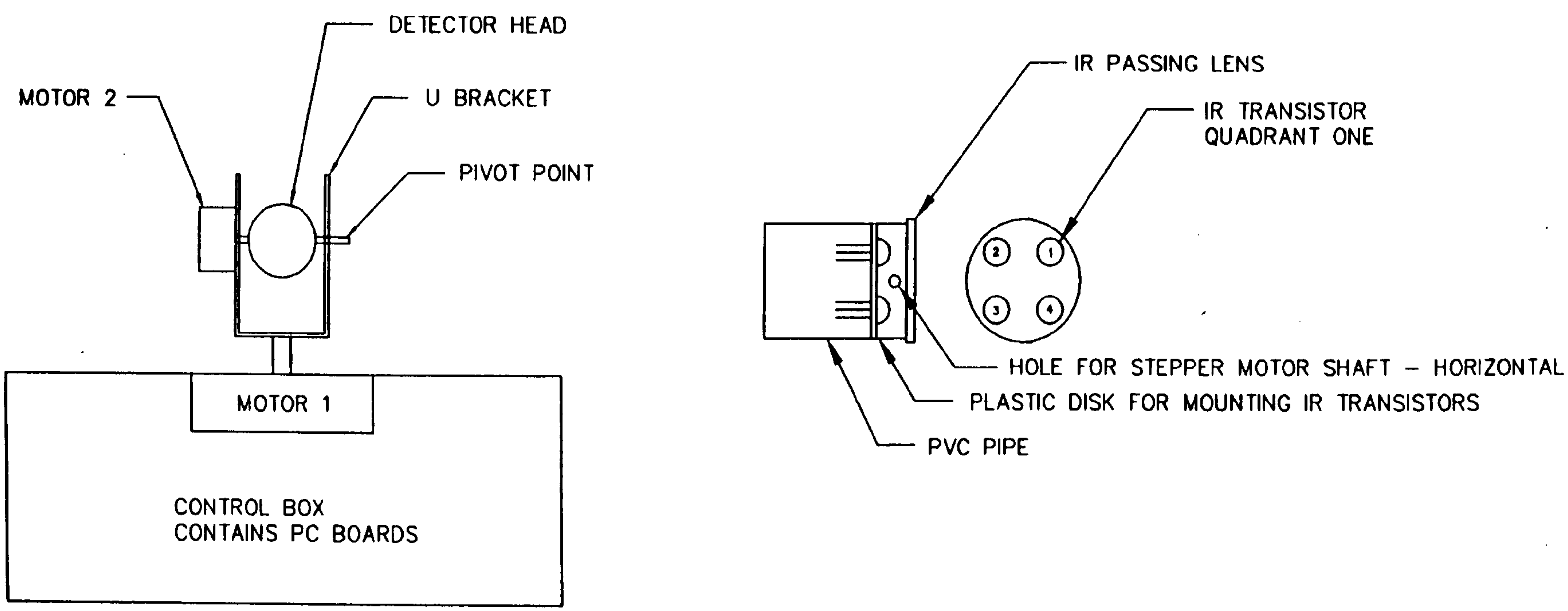


FIGURE 4. Detective Head and Tracking Mechanism.

TABLE 2: TRACKING OPTIONS

QUADRANT 1 IS GREATEST	**MOTOR 1 CCW MOTOR 2 CW**
QUADRANT 2 IS GREATEST	**MOTOR 1 CW MOTOR 2 CW**
QUADRANT 3 IS GREATEST	**MOTOR 1 CW MOTOR 2 CCW**
QUADRANT 4 IS GREATEST	**MOTOR 1 CCW MOTOR 2 CCW**
QUADRANT 1 & 2 EQUAL AND GREATEST	**MOTOR 1 NO MOVEMENT MOTOR 2 CW**
QUADRANT 2 & 3 EQUAL AND GREATEST	**MOTOR 1 CW MOTOR 2 NO MOVEMENT**
QUADRANT 3 & 4 EQUAL AND GREATEST	**MOTOR 1 NO MOVEMENT MOTOR 2 CCW**
QUADRANT 1 & 4 EQUAL AND GREATEST	**MOTOR 1 CCW MOTOR 2 NO MOVEMENT**
ALL QUADRANTS EQUAL	**MOTOR 1 NO MOVEMENT MOTOR 2 NO MOVEMENT**

Pragramming for the STAMP II

'ASSIGNMENT OF VARIABLES AND CONSTANTS'

```
LIN_100: SENSE          VAR  BYTE
LIN_110: SENSE_1        VAR  BYTE          ‘INPUTS FROM THE
                                       A/D CONVERTERS ARE’
LIN_120: SENSE_2        VAR  BYTE          ‘ASSIGNED TO
                                 THESE  VARIABLES WITH’
LIN_130: SENSE_3        VAR  BYTE          ‘8 BIT RESOLUTION’
LIN_140: SENSE_4        VAR  BYTE

LIN_150: HIGH1_1        VAR  SENSE_1.HIGHNIB     ‘SET HIGH1_?
```

```
                                                    EQUAL TO THE UPPER'
LIN_160: HIGH1_2           VAR  SENSE_2.HIGHNIB     '4 BITS OF THE
                                                         CORRESPONDING'
LIN_170: HIGH1_3           VAR  SENSE_3.HIGHNIB     'SENSE_?
                                                         VARIABLE'
LIN_180: HIGH1_4           VAR  SENSE_4.HIGHNIB

LIN_190: ERROR_1           VAR  HIGH1_1.BIT0        'ERROR_? SET
                                                         QUAL TO THE LSB'
LIN_200: ERROR_2           VAR  HIGH1_2.BIT0        'OF THE
                                                    HIGH1_?  VARIABLE'
LIN_210: ERROR_3           VAR  HIGH1_3.BIT0
LIN_220: ERROR_4           VAR  HIGH1_4.BIT0

LIN_230: HIGH_1            VAR  NIB            'WILL REPRESENT UPPER
                                               4 BITS OF THE'
LIN_240: HIGH_2            VAR  NIB            'CORRESPONDING
                                          SENSE_? VARIABLE'
LIN_250: HIGH_3            VAR  NIB            'AFTER CONDITIONING'
LIN_260: HIGH_4            VAR  NIB

LIN_270: OFFSET            VAR  BYTE                'INPUTS FROM THE
                                               A/D CONVERTERS ONLY'
LIN_280: OFFSET_1          VAR  BYTE                'DURING THE
                                AMBIENT SENSOR READING'
LIN_290: OFFSET_2          VAR  BYTE                'SECTION USED TO
                                          NULIFY AMBIENT LIGHT'
LIN_300: OFFSET_3          VAR  BYTE
LIN_310: OFFSET_4          VAR  BYTE

LIN_320: CS                 VAR NIB            'CHIP SELECT LINE USED
                                               W/ A/D'
LIN_330: ADC                CON 13             'DATA LINE USED W/ A/D
                                               - PIN 13'
LIN_340: CLK                CON 12             'CLK LINE USED W/ A/D -
                                               PIN 12'
LIN_350: ADCMIN            CON  50             'MIN ACCEPTABLE A/D
                                                    OUTPUT'
```

```
LIN_360: MOVE           VAR  NIB      'STEPPED TO DECIDE
                                       WHICH DATA WORDS'
LIN_370: MOVE2          VAR  NIB      'ARE USED TO STEP
                                  MOTORS 1& 2'

LIN_380: VALUE          VAR  NIB      'THE VALUES OF THE
                                       DATA WORDS USED'
LIN_390: VALUE2         VAR  NIB      'TO STEP MOTORS 1 & 2'

LIN_410: JUMP1          VAR  BIT      'USED TO CONTROL CW &
                                       CCW ROTATION'
LIN_420: JUMP2          VAR  BIT      'OF MOTORS 1 & 2'
LIN_430: JUMP3          VAR  BIT

LIN_440: MOVE = 1: MOVE2 = 1          'ASSIGNS A STARTING
                                       POINT'
LIN_450: DIRS = %1101111111110000     'SETS PINS TO INPUT OR
                                       OUTPUT'

'END OF ASSIGNMENTS'

'DISABLE ALL A/D CONVERTERS'

LIN_460: FOR CS = 0 TO 3
LIN_470: HIGH CS               'SET CHIP SELECT LINES HIGH TO
                                  DISABLE'
LIN_480: NEXT

'END A/D DISABLE'

'TAKE AMBIENT SENSOR READING'

LIN_500: FOR CS = 0 TO 3
LIN_510: LOW CS                   'IMPORTING ADC DATA FROM
                                  SENSORS'
LIN_520: SHIFTIN ADC,CLK,MSBPOST,[OFFSET\9]
LIN_530: HIGH CS
```

```
LIN_540: IF CS = 0 THEN LIN_580
LIN_550: IF CS = 1 THEN LIN_590
LIN_560: IF CS = 2 THEN LIN_600
LIN_570: IF CS = 3 THEN LIN_610

LIN_580: OFFSET_1 = OFFSET: GOTO LIN_620     'ASSIGN INDIVIDUAL
                                             AMBIENT VALUES'
LIN_590: OFFSET_2 = OFFSET: GOTO LIN_620     'TO AN OFFSET
VARIABLE'
LIN_600: OFFSET_3 = OFFSET: GOTO LIN_620
LIN_610: OFFSET_4 = OFFSET
LIN_620: NEXT
LIN_630: GOTO LIN_705

'END READING'

'POLE THE SENSORS'

LIN_700: HIGH 15: LOW 15                  'ACTIVATE LOCK ON LED'
LIN_705: FOR CS = 0 TO 3
LIN_708: HIGH 14: LOW 14                  'ACTIVATE POLING LED''
LIN_710: LOW CS
LIN_720: SHIFTIN ADC,CLK,MSBPOST,[SENSE\9]
LIN_730: HIGH CS

LIN_740: IF CS = 0 THEN LIN_800
LIN_750: IF CS = 1 THEN LIN_830
LIN_760: IF CS = 2 THEN LIN_860
LIN_770: IF CS = 3 THEN LIN_890

LIN_800: IF SENSE > ADCMIN THEN LIN_820              'ASSIGN
                                                 INPUT VALUE'
LIN_810: SENSE = OFFSET_1                        'FROM A/D
                                                 CONVERTERS'
LIN_820: SENSE_1 = SENSE - OFFSET_1: GOTO LIN_920    'AFTER
                                                 SUBTRACTING'
                                                 'THE AMBIENT
                                                 OFFSET'
```

```
LIN_830: IF SENSE > ADCMIN THEN LIN_850                    'VALUES'
LIN_840: SENSE = OFFSET_2
LIN_850: SENSE_2 = SENSE - OFFSET_2: GOTO LIN_920

LIN_860: IF SENSE > ADCMIN THEN LIN_880
LIN_870: SENSE = OFFSET_3
LIN_880: SENSE_3 = SENSE - OFFSET_3: GOTO LIN_920

LIN_890: IF SENSE > ADCMIN THEN LIN_910
LIN_900: SENSE = OFFSET_4
LIN_910: SENSE_4 = SENSE - OFFSET_4

LIN_920: NEXT

'END SENSOR POLING'

'DATA CONDITIONING'

LIN_930: HIGH_1 = HIGH1_1:HIGH_2 = HIGH1_2:HIGH_3 =
HIGH1_3:HIGH_4 = HIGH1_4

LIN_940: IF ERROR_1 = 1 THEN LIN_960     'MAKE BIT 0 OF HIGH_?
                                          ALWAYS'
LIN_950: HIGH_1 = HIGH1_1 + 1        'A 1  XXX1 & XXX0 = XXX1'
LIN_960: IF ERROR_2 = 1 THEN LIN_980
LIN_970: HIGH_2 = HIGH1_2 + 1
LIN_980: IF ERROR_3 = 1 THEN LIN_1000
LIN_990: HIGH_3 = HIGH1_3 + 1
LIN_1000: IF ERROR_4 = 1 THEN LIN_1170
LIN_1010:HIGH_4 = HIGH1_4 + 1

'END CONDITIONING'

'COMPARE ALL SENSOR READINGS TO MIN ACCEPTABLE VALUE'

LIN_1170: IF SENSE_1 < ADCMIN AND SENSE_2 < ADCMIN AND
SENSE_3 < ADCMIN AND SENSE_4 < ADCMIN THEN LIN_705
```

'END COMPARISON'

'DETERMINE TRACKING ACTIONS'

LIN_1200: IF HIGH_1 > HIGH_2 AND HIGH_1 > HIGH_3 AND HIGH_1 > HIGH_4 THEN LIN_2000
LIN_1210: IF HIGH_2 > HIGH_1 AND HIGH_2 > HIGH_3 AND HIGH_2 > HIGH_4 THEN LIN_2020
LIN_1220: IF HIGH_3 > HIGH_1 AND HIGH_3 > HIGH_2 AND HIGH_3 > HIGH_4 THEN LIN_2040
LIN_1230: IF HIGH_4 > HIGH_1 AND HIGH_4 > HIGH_2 AND HIGH_4 > HIGH_3 THEN LIN_2060
LIN_1240: IF HIGH_1 = HIGH_2 AND HIGH_1 > HIGH_4 AND HIGH_1 > HIGH_3 THEN LIN_2080
LIN_1250: IF HIGH_3 = HIGH_4 AND HIGH_3 > HIGH_1 AND HIGH_3 > HIGH_2 THEN LIN_2100
LIN_1260: IF HIGH_2 = HIGH_3 AND HIGH_2 > HIGH_1 AND HIGH_2 > HIGH_4 THEN LIN_2120
LIN_1270: IF HIGH_1 = HIGH_4 AND HIGH_1 = HIGH_2 AND HIGH_1 > HIGH_3 THEN LIN_2140
LIN_1280: IF HIGH_1 = HIGH_2 AND HIGH_2 = HIGH_3 AND HIGH_3 = HIGH_4 THEN LIN_700

'END DETERMINATION'

'TRACKING ACTIONS'

LIN_2000: JUMP1 = 1: JUMP2 = 0: JUMP3 = 0: GOTO LIN_3100
LIN_2020: JUMP1 = 0: JUMP2 = 0: JUMP3 = 0: GOTO LIN_3100
LIN_2040: JUMP1 = 0: JUMP2 = 1: JUMP3 = 0: GOTO LIN_3100
LIN_2060: JUMP1 = 1: JUMP2 = 1: JUMP3 = 0: GOTO LIN_3100
LIN_2080: JUMP2 = 0: GOTO LIN_4100
LIN_2100: JUMP2 = 1: GOTO LIN_4100
LIN_2120: JUMP1 = 0: JUMP3 = 1: GOTO LIN_3100
LIN_2140: JUMP1 = 1: JUMP3 = 1: GOTO LIN_3100

'END TRACKING ACTIONS'

'STEPPING ROUTINES'

'CW MOTOR1'

```
LIN_3100: IF JUMP1 = 1 THEN LIN_3400
LIN_3120: IF MOVE = 7 THEN LIN_3300
LIN_3140: MOVE = MOVE + 1
LIN_3160: LOOKUP MOVE,[1,3,2,6,4,12,8,9],VALUE
LIN_3180: OUTB = VALUE
LIN_3200: IF JUMP3 = 1 THEN LIN_705
LIN_3220: GOTO LIN_4100

LIN_3300: MOVE = MOVE - 8: GOTO LIN_3140
```

'CCW MOTOR1'

```
LIN_3400: IF MOVE = 0 THEN LIN_3510
LIN_3420: MOVE = MOVE - 1
LIN_3440: LOOKUP MOVE,[1,3,2,6,4,12,8,9],VALUE
LIN_3460: OUTB = VALUE
LIN_3480: IF JUMP3 = 1 THEN LIN_705
LIN_3500: GOTO LIN_4100

LIN_3510: MOVE = MOVE + 8: GOTO LIN_3420
```

'CW MOTOR2'

```
LIN_4100: IF JUMP2 = 1 THEN LIN_5100
LIN_4120: IF MOVE2 = 7 THEN LIN_4300
LIN_4140: MOVE2 = MOVE2 + 1
LIN_4160: LOOKUP MOVE2,[1,3,2,6,4,12,8,9],VALUE2
LIN_4180: OUTC = VALUE2
LIN_4200: GOTO LIN_705

LIN_4300: MOVE2 = MOVE2 - 8: GOTO LIN_4140
```

'CCW MOTOR2'

```
LIN_5100: IF MOVE2 = 0 THEN LIN_5300
LIN_5120: MOVE2 = MOVE2 - 1
LIN_5140: LOOKUP MOVE2,[1,3,2,6,4,12,8,9],VALUE2
LIN_5160: OUTC = VALUE2
LIN_5180: GOTO LIN_705

LIN_5300: MOVE2 = MOVE2 + 8: GOTO LIN_5120

'END STEPPER ROUTINES'
```

BIOGRAPHICAL INFORMATION

Mohammad Fotouhi is an associate professor of Electrical Engineering Technology at the University of Maryland Eastern Shore. He received his Ph.D. in Power System Engineering from the University of Missouri-Rolla, M.S. from Oklahoma State University and B.S. from Tehran Polytechnic College. He has been conducting a practical research on growth and characterization of diluted magnetic semiconductors since 1986. He is a member of Etta Kappa Nu Honor Society and IEEE. He was Chairman of Student and Industry Relations and Host Committee Member of IEEE Conferences on: Power System Computer Application (PICA '91), and Power Engineering Society, Winter Meeting, (PES '96) both held in Baltimore, Maryland.

William Cavey began his education by completing an Associate in Arts degree in Electronics Engineering Technology from Delaware Technical Community College. He then went on to complete his Bachelor of Science degree in Electronics Engineering Technology at the University of Maryland Eastern Shore. He is enrolled as a graduate student with NUT University to obtain a Master of Science degree in Electrical Engineering. Currently William works at K & L Microwave, Inc. as a RF/Microwave Design Engineer.