MIGUEL ANGEL ARNEDO
SPRING 2001
Date: 20/4/01
Instructor: A. Arroyo
TA: Scott Nortman
Rand Candler

University of Florida
Department of Electrical and Computer Engineering
EEL 566
Intelligent Machines Design Laboratory

MIL FINAL WRITTEN REPORT

TABLE OF CONTENTS:

1.ABSTRACT

The following document discusses the design of an autonomous robot. This robot should be able to react to different situations in his environment. It will be equipped with several sensors in order to read what is happening in the outside world. The brain is going to be a HC11 microprocessor.

2.INTRODUCTION. GOAL. SPECIFICATIONS.

The goal of my project is to build a robot that can have two different behaviors: be your friend or attack anything that is moving around his area.

a) Friend behavior: The robot will look for human presence around him. As soon as he realizes that somebody is close by, he will detect his exact position and will go where that person is. Then, he will follow the fellow wherever he goes.

b) Enemy behavior: The robot will shoot a disc to anything (objects or people) that is moving around him. He will detect his exact position and then he will shoot him with a disc gun. These discs will be launched when the robot has been orientated towards to the target.

Other specifications are: Wireless communication with the PC and a videocamera system.

The PC will send to the robot the operation mode in each time (friendly or non-friendly). Also, can adjust the velocity of the robot motion.

The video camera will be connected wireless to a TV , where it will be possible to see the person that the robot is following or the enemy that is attacking, accordingly with the mode of operation in that time. The video camera will turn off as soon as there is no enough light in the room.

The robot will move randomly across the room until detects the presence of a human. In that point, will follow the human, or shoot him, depending on the mode of operation of the robot in that moment. Should be able to avoid collisions with objects in his path.

The *main objective* of the project is to localize in the space as best as possible the people and objects moving around the robot. If there is no good accuracy, it is going to be hard for the robot to reach the person or to shoot a disc to an object that is moving.

The following sections discuss each system of the robot. The report starts with the integrated system, which will describe the role of all the components used. The next section will explain the basic characteristics of the base of the robot, the platform. Actuation section discusses how the robot achieves its motion. The sensor section describes how the robot is going to communicate with his surrounding. The next section goes in how every behavior is achieved, followed by the wireless communication system. The power requirements for the robots are studied in the next part. The budget, conclusion and lesson learned, the possible improvements of the robot in the future and the software are the last sections.

3. INTEGRATED SYSTEM

The robot brain will be a Motorola 68Hc11 connected to the TJ-PRO board. The software will check the sensors, and the routines will make decisions based on the sampled values.

The schematic of this board is included in the next pages. This information is detailed in the 'TJTM Pro Assembly Manual', in the WEB: mekatronix.com/manuals/manuals.htm

Characteristics of MTJPRO11. Memory and IO mapping:

- 32 Kbytes of SRAM. From 0x8000 to 0xFFFF.
- 8 digital outputs.
- 3 digital input ports that can be used as input capture.
- Provides 4 active-low input and 4 active-low output latch enables, which are memory mapped in the following addresses:

Table 1 Memory Map of MTJPRO11™ IO Enables

Name	Direction	Memory Address (Hex)
Y0	Output	0x4000
Y1	Input	0x4000
Y2	Output	0x5000
Y3	Input	0x5000
Y4	Output	0x6000
Y5	Input	0x6000
Y6	Output	0x7000
Y7	Input	0x7000

- Five 8 bits analog input channels. Can be combined with external analog multiplexers and digital input address decoding, allowing each analog channel to be multiplexed to as many channels as the external multiplexer allows.
- Hardware generation of 40KHz signal to modulate IR and sonar.

To develop the application we require communication between the PC and the TJ pro. This task will be done through the M2325 bidirectional communication board.
TJ pro programs can be written in assembly, basic or C. I will use the C language. The software used is called HC11 (Interactive C Compiler). The program will be downloaded into the microprocessor through the serial port of both PC and HC11.

In order to meet the specifications, the system will be divided in these main groups:

3.1. Power supply system
3.2. Sensors
3.3. Actuators
3.4. Wireless communication with the PC
3.5. Videocamera system.

All these groups will be discussed below.

In addition, the system is equipped with:

- Switches:
  a) Reset push button
  b) Toggle switch: Down load or run the program in the TJ pro.
  c) Toggle switch: Turns On-Off the robot

- Led of 'power on'.

A complete graphical description of the system is included in next pages.

4. MOBILE PLATFORM

The robot platform was built over a toy platform. Some adjustments was required.

Some parts of the platform were designed with Auto-Cad. These parts were cutted with a 'T-Tec machine'. Mainly are the base of the printed circuit boards and the little platform where the video camera is sited. The autocad design is included in the next pages.

The switches and leds discussed in the 'integrated system' section of this document were placed on the upper part of the platform.

The circuits are down the cover of the platform. The 3 battery packs, two outside of the robot, and the third one, inside, on the bottom of the robot.

The objectives: should be big enough to place all the devices that the robot includes. Also should be robust. The lighter the better. A good distribution of the weight for a good mobility.

5. ACTUATION:

This section discusses the actuators included in the robot. Can be divided into three functional groups. This is discussed in the three points below.

5.1. Mobility of the robot:

The mobility of the robot is done with motors. These motors were included in the toy that I bought as platform. In order to control them, a printed circuit board controls their velocity. These motors are not perfectly equal. Therefore accommodations had to be made to get them match up, via software, accordingly with what the shaft encoders dictate.

These motors will allow the robot forward, backward and turn movements.

5.2. Shooting Disc Gun Machine:

In order to do the shots, the robot is equipped with a disc gun. This system requires two motors connected to this gun in order to launch the discs. One motor makes the disc machine to be on/off. The other will perform the trigger.

5.3. Rotation of the video camera:

The video camera is installed above a small circular platform, which can rotate. This platform is connected through mechanically to a stepper motor. By doing that, we can know the exactly position at where the video camera is pointing at.

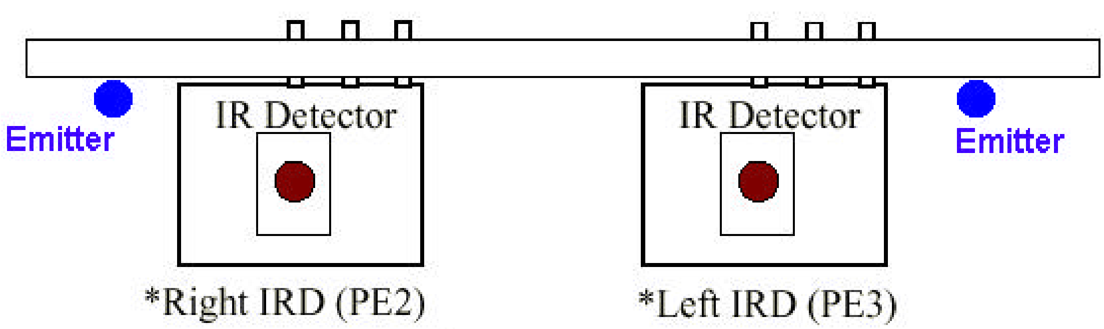
The objectives in this area are to keep a good relationship between torque and velocity.

6. SENSORS:

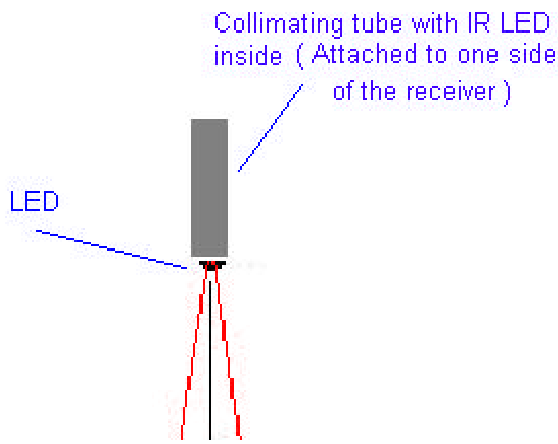
The robot will be equipped with mainly five functional groups of sensors: to avoid collisions, to detect motion, to improve the navigation system, to detect the lack of light in the room and to detect any noise. The first two groups will be placed around the robot, in strategic positions according the angle of vision of each sensor. The third group is connected to the wheels, in order to count how many rpm has each wheel.

6.1 Sensors used to avoid collisions:

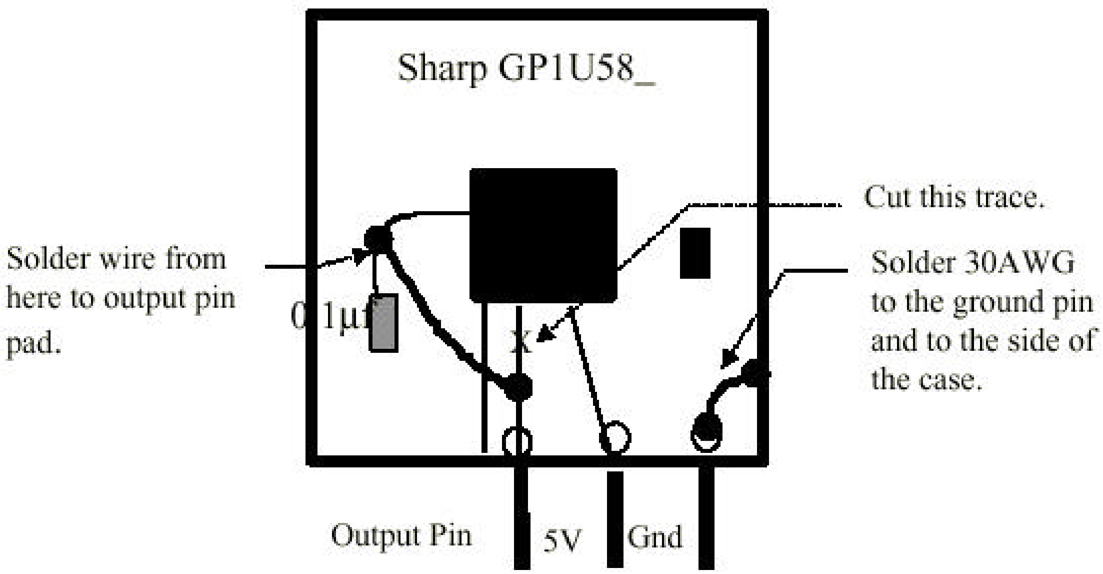
- The IR sensors will avoid collisions against the walls or another objects close to the robot. These infrared sensors work at 40KHz, with a wavelength of 940 nm. The robot is equipped with two forward-looking IR Emitters and two IR receptors.



- The Emitters are Infrared LED, placed inside of a collimating tube.



- The Receivers are GPIU58Y hacked sensors. I hacked the Receivers following the next steps:



The analog output voltage of the Receiver varies from 1.5 to 2.5 Volts with a rise time of 100 ms and a fall time of 50ms. The ADC will typically provide digital outputs in the range of 88 to 130, yielding a range about 5 bits of precision.

The effective range of the IR sensor depends upon the IR emitter illumination level and the degree of beam collimation. With a current of 5 mA though uncollimated IR emitters, the effective ranges varies from about 4 inches to 16 inches, ideal for proximity sensing.

The next table shows the values measured for the IR receivers. Note: due to the platform dimensions and the IR calibration, the thresholds of both IR receivers have no identical values.

Analog values measured in the IR receivers

	LEFT IR	**RIGHT IR**
With no obstacle	85	90
Aprox. Threshold	89	95

The left IR is in PE3 pin.
The right IR in PE2 pin.

- The micro-switches will help the robot to get away from those objects if there was a collision. Are mounted around the perimeter of the robot. If the robot runs into any obstacles the switches will be pressed. Each one is separately identificable via hardware, due to the fact that every one gives different analog voltages.

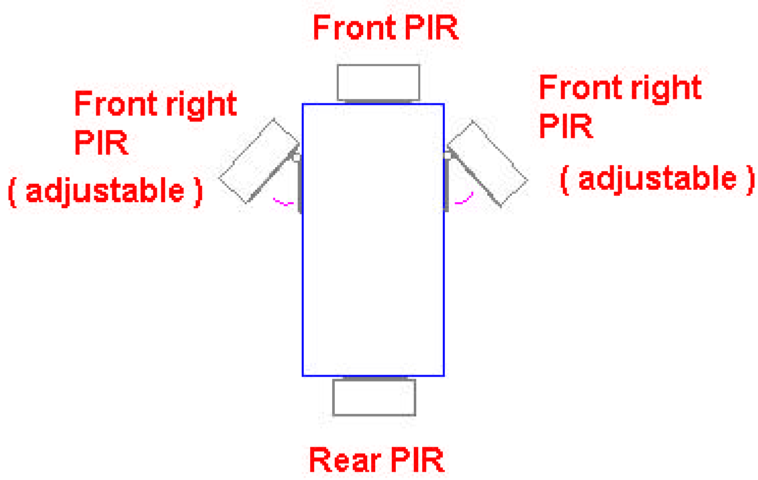
Analog values measured for the bumpers

Bumper	**Analog value**
Front left	20..22
Rear	78..79
Front right	125..128

6.2. Sensors used to detect motion:

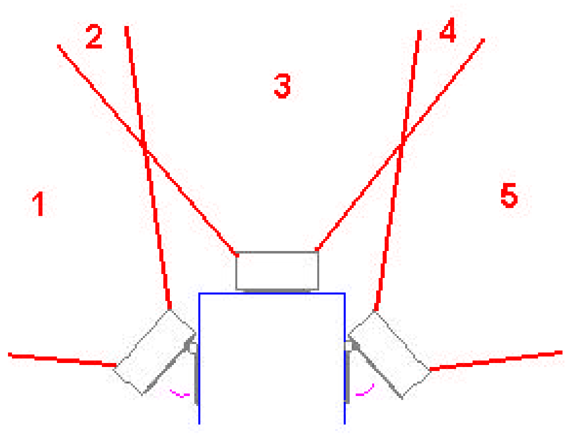
The motion detectors will be pyrodetectors to detect the heat of the human presence. These IR works at 32KHz. Their situation in the platform is shown in the next figure:

Figure 6.2.1. Position of the PIRS in the robot.



In order to localize with exactness the position of the heat source (the human), the PIR are placed in a strategic position. The 3 front PIR sweep 5 different areas.

Figure 6.2.2. Different areas distinguish by the PIRS of the robot.



For example, if the Hc11 receive signals from both the front left and front PIRs, the robot will assume that the human is on the front left.

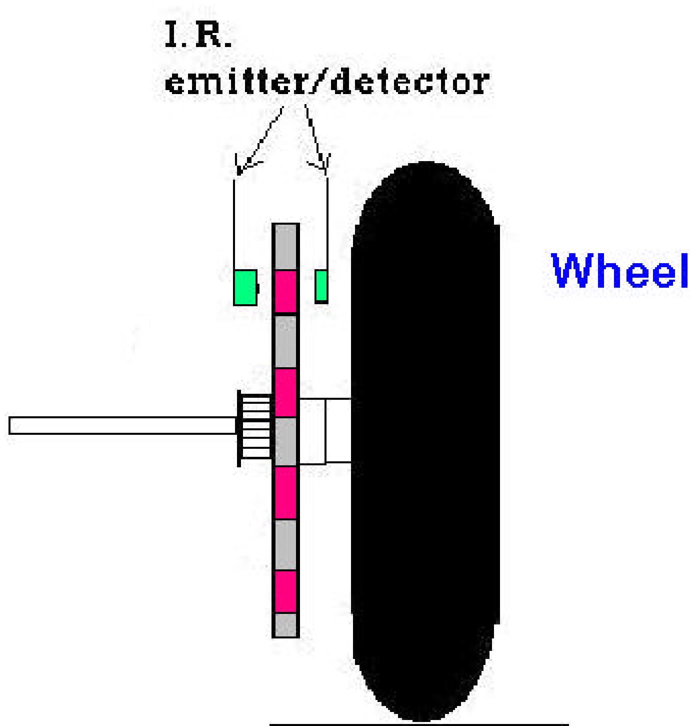
The technical specifications of the PIR modules used are exposed in the next pages. It is called KC7783 module, and uses a KC778B control chip.

The sensors only work in indoors. The sensors are quite sensitive, so are read when the robot is stopped.

6.3. Navigation system:

The direction of movement along the room will be randomly. A navigation system makes sure that the robot is following a straight path when desired. It consists of a shaft encoding system.

As shown in the next picture, a pair of emitter detector LED and a stripped wheel are used to perform this task.



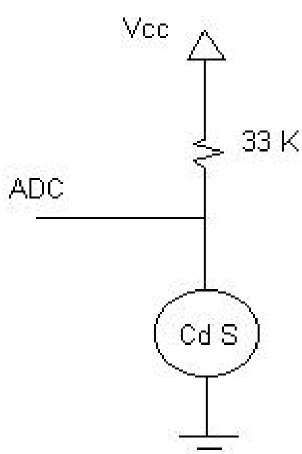
The stripped wheel is placed in the shaft of the wheel. The emitter is placed in one side of the stripped wheel. The receiver is in the opposite side.

The width of the stripped wheels an important issue in designing encoders. Although more stripes give greater resolution to the measurements, the stripes can not be smaller than the field of view of the photodetector.

6.4. The light sensor

The robot needs to detect the lack of light in the room. As stated before, as soon as the light is not enough to make the video camera work properly, this will be disconnected.

The circuit that makes this task possible is quite simple.



Analog values measured for the light detector

Enviroment	**Analog value**
No light enough	> 200
Room light	< 200

6.5. *The noise detector:*

Due to the fact that these detectors are not very sensitive, the sensor should be cover with a cone in order to direct the sensor to some point. Also, the great amount of noise done by the motors and the robot moving, the noise is only detected if the robot is completely stopped. I am considering to include a band pass filter in order to avoid noises from the motors. The schematic of the circuit is on the page 143 of the book ‘Mobile robots, an Inspiration to implementation.

Analog values measured for the noise detector

Sound	**Analog value**
No sound	127…130
Clapping, load voice	Other

7.BEHAVIORS

The main behaviors that the robot have are:

- Obstacle avoidance
- Bump sensing
- Human following
- Target finding and triggering
- Vision
- Receiving orders from the PC

7.1. Obstacle avoidance:

. The robot moves randomly across the room. In any moment can crash into unexpected obstacles. Based on the readings of the IR sensors placed in front of the robot, this is able to avoid collisions with obstacles in his environment

Based on experiments, I fixed a threshold point. If the robot receives information from any of the IR sensor above this value, it means that an object is in front of him.

If only the left front IR detects any obstacle, the robot will turn right. If only the right front IR detects any obstacle, will turn left. If both detect obstacles, the robot will go back, and then, will turn.

7.2. Bump sensing

Two bumper switches are strategically placed in the front of the robot. If the IR's fail to detect obstacles in front of the robot, the left front, the right front, or both switches will be pressed. The robot will go back accordingly in order to get separated from the obstacle.

7.3.Human following. Friendly behavior.

This behavior is performed in the friendly mode.
As soon as anybody is detected by the PIR sensors, the robot will turn until the front PIR detect the human. In that moment, the robot will go straight forward to him. Due to the fact that the PIRs does not work properly when the robot is moving, every x seconds the robot stops and reads the PIR to make sure that the robot is going in the appropiate direction. If it does not, just turn to the left or the right, until the front sensor states again that the person is in front of the robot. Then, the robot will keep on going forwards, until the IR and/or the sensors are pressed. In that moment the human is considered reached.

The next table shows the different states of the robot in this mode of operation. Each state corresponds to a different action. Every mode requires a certain time until the robot changes to another action.

The time is controlled using the Time Overflow feature of the HC11. Programming the preescaler appropriately, we can obtain an interrupt every 524.3 ms.. So, approximately, we need almost 2 ints to obtain 1 sec. See the program for more details.

State of the robot (Action)	Time in that mode
(1)GO_FORWARDS	TIME_LEFT_UNTIL_CHANGE_DIRECTION
(2)GO_BACKWARDS	TIME_GOING_BACKWARDS
(3)GO_FORW_TURN_RIGHT	TIME_IN_TURN_RIGHT
(4)GO_FORW_TURN_LEFT	TIME_IN_TURN_LEFT
(5)GO_FORW_TURN_RIGHT_LITTLE	TIME_IN_TURN_LITTLE_RIGHT
(6)GO_FORW_TURN_LEFT_LITTLE	TIME_IN_TURN_LITTLE_LEFT
(7)GO_BACK_TURNING_RIGHT	TIME_TURNING_WHEN_BUMPER_HIT
(8)GO_BACK_TURNING_LEFT	TIME_TURNING_WHEN_BUMPER_HIT
(9)CHOOSE_ANOTHER_DIRECTION	TIME_BASE_FOR_RANDOM_NUMBER x Random #
(10)ESTABILIZATING_PIRS	TIME_TO_ESTABILIZATE_PIRS
(11)STOP_FOR_SCANNING	TIME_SHOULD_KEEP_STOPPED_WHEN_SCAN

7.3.1. Go_forwards: The robot goes forwards.

If the robot senses any obstacle in front of him, by the bumpers and/or IR detectors, changes his direction in order to avoid the obstacle. The robot changes his current status (1), to another states. These states could be (2),(7) or (8) in the case of bumping frontally, rightly or lefty, or (2) (5) or (6) in the case of IR detectors (both, left or right IR, respectively).

If the robot does not detect any obstacle in his path, and he has spent already a time greater than TIME_LEFT_UNTIL_CHANGE_DIRECTION, the robot changes to (9) status.

7.3.2. Go_backguards: In the case of bumping or IR sensing of a obstacle in front of the robot.

7.3.3. Go_forw_turn_right: A human has been detected in the area 5. The robot turns right in order to look forwards to him. See Figure 6.2.2. Different areas distinguish by the PIRS of the robot.

7.3.4. Go_forw_turn_left: Idem for the area 1, turning left.

7.3.5. Go_forw_turn_right_little: Idem for area 4, turning slightly to the right.

7.3.6. Go_forw_turn_left_little: Idem for area 2, turning slightly to the left.

7.3.7. Go_back_turning_right: See 7.3.1.

7.3.8. Go_back_turning_left: See 7.3.1.

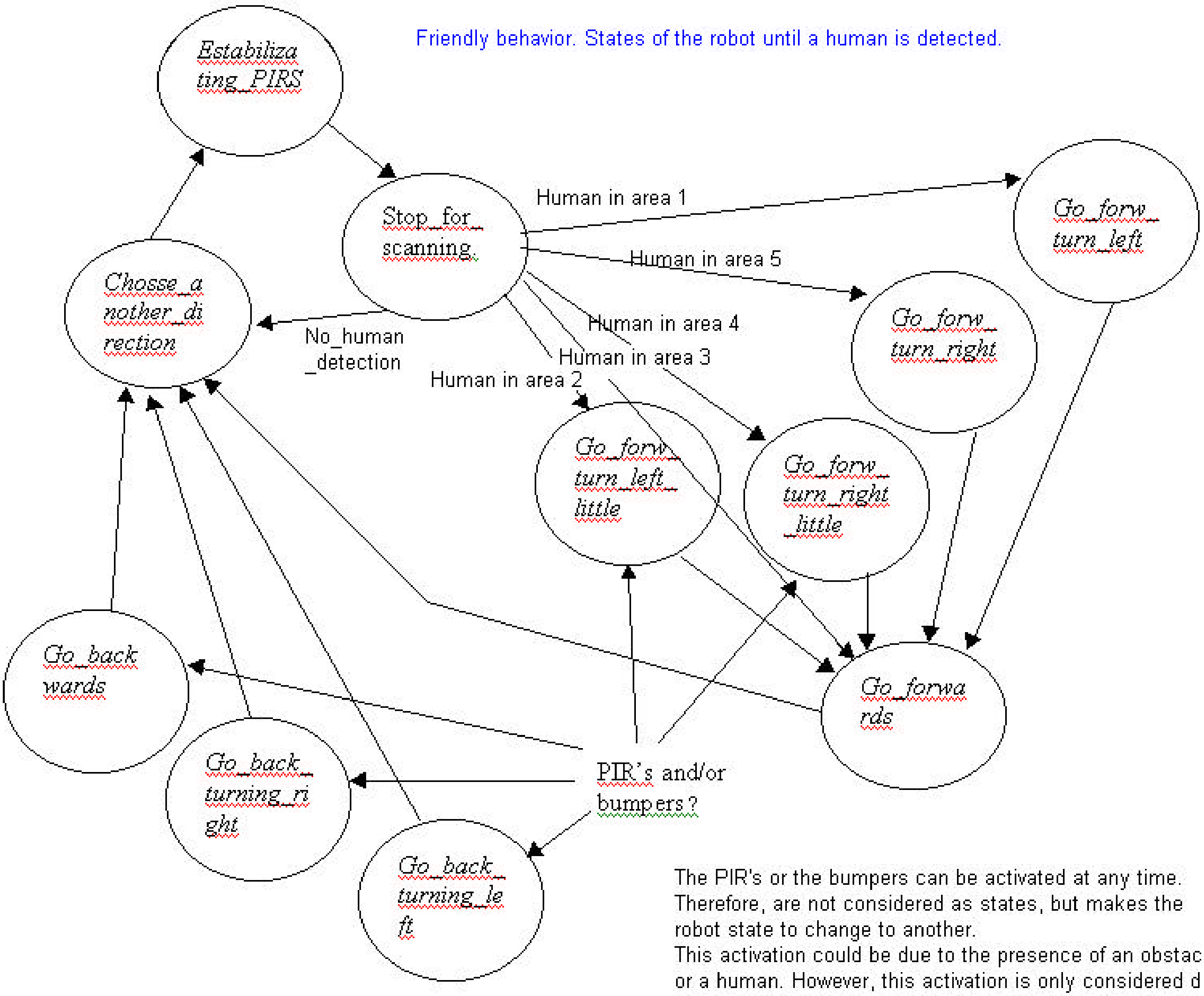
7.3.9. Chosse_another_direction: The robot did not find any body in his path. Will turn randomly to start looking for somebody in other direction.

7.3.10. Estabilizating_PIRS: The PIRS are very sensitive. The robot should be completely stopped before doing any reading on the PIRS. After any movement of the robot, the robot will be stopped, doing nothing, waiting for the PIRS to be estabilizated.

7.3.11. Stop_for_scanning: Ones the PIRS are estabilizated, the robot can scan if there is somebody around him.

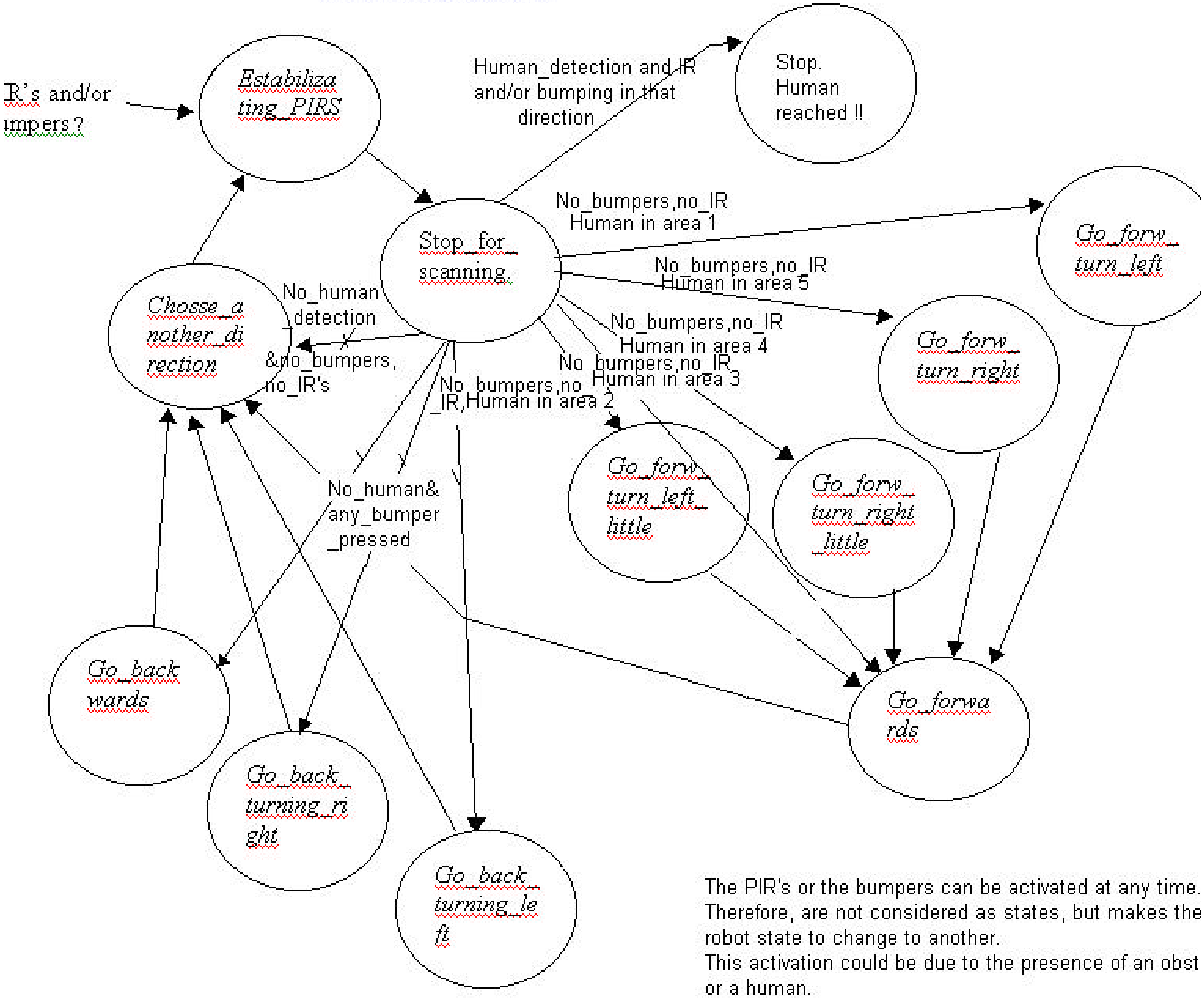
The different states that the robot has in the friendly behavior are graphically exposed next. The first graphic represents the different status of the robot until a human presence is detected. The second, the status ones is detected until the human is reached.

Friendly behavior. States of the robot until a human is detected.



The PIR's or the bumpers can be activated at any time. Therefore, are not considered as states, but makes the robot state to change to another.
This activation could be due to the presence of an obstac or a human. However, this activation is only considered d to an obstacle, until 'stop for scanning' status detect any human. See next flow chart in that case.

Friendly behaviour. States of the robot since a human is detected until is reached.



The PIR's or the bumpers can be activated at any time. Therefore, are not considered as states, but makes the robot state to change to another.
This activation could be due to the presence of an obst or a human.

Now both cases are considered.

7.4.Target finding and triggering. Enemy behavior.

This is performed when the robot is operating in the enemy behavior. The robot will be looking randomly across the room. The intruder is localized with the PIR's. Then, the robot turns until the enemy is in front of him, activating his gun and shooting him.

In addition of the different actions in the friendly behavior, now the robot should activate his gun and shoot the disc.

State of the robot (Action)	**Time in that mode**
SHOOTING	TIME_TO_PRODUCE_ONE_SHOOT
END_SHOOTING	TIME_TO_PRODUCE_ONE_SHOOT

7.4.1. Shooting: the robot has the enemy in front of him. Activates the gun and triggers the disc.

7.4.2. End_shooting: the robot has done the shoot. Now turns off the gun and recovers the trigger to its initial position.

7.5.Vision

The robot accomplishes a survey task. That's why is important to have a video system. The video camera will transmit the images wireless to the receiver, attached to any TV or PC with video card. The video camera will stop recording as soon as the light sensor detects that there is no light enough in the room.

7.6.Receiving orders from the PC

The PC sends orders to the HC11 in the robot via wireless communication. Such orders can be: velocity of robot's motion, friendly or enemy modes of operation. This behavior is not implemented yet, because of some problems with the wireless communication.

8. WIRELESS COMMUNICATION SYSTEM

The wireless communication makes possible to send orders from the PC to the robot. This system consists of a transmitter and a receiver. The transmitter is connected to the parallel port of the PC. The receiver, to 4 input pins of the HC11. This 4 pins are read by polling.

It's a 4 bit transmission system. This system is not working YET, so I will not include more details.

9.POWER REQUIREMENTS. BATTERIES.

- The boards require 6 AA Nickel Cadmium rechargeable batteries with at least 600 ma-hr capacity, 5.4-7.2 volts. Only Nickel Cadmium batteries are recommended for the board.

- The video camera and the boards that control the motors require 12 V.

Then, one 6AA battery pack will supply the TJPRO board with the HC11, and two 6AA battery pack connected in serie will supply two regulators:

- 5 V regulator: supplies the PIR's and the receiver transmission board.
- 12 V regulator: supplies the motor boards and the video camera.

Each pack of 6 AA batteries supplies 6x1.2=9 volts when the batteries are fully charged.

10. BUDGET

Concept	Cost
MTJPRO11 microp + expansion board	65
MB2325 communication board	15
IR detectors	2x3
IR Leds	2x0.75
SW DPST	1.5
SW SPST	1.5
SW Reset	1.5
Wire wrapping tool	7
Batteries	18 x 2
Wireless RX/TX + post	35
Perf boards	14
Lever Arm Switches	3 x 2
Platform	2 x 20
X10 Video camera + Post	90
PIR detector KC7783	4 x 9.5
Quad comparators	3 x 1
Regulators	2 x 1.49
	Total:…………. 354.48 $

Other: resistors, capacitors, CD S cellule, Microphone, Wires, male and female headers, microswitches, leds, tools and boards were supplied by the MIL.

11. CONCLUSION. LESSON LEARNED.

The PIR's are very sensitive, so the human is only detected when the robot is stopped. Also, the motors are very noisy so makes the noise detector be useful only when the robot is stopped. The video camera and the motor boards require 12 volts, which makes necessary to have many batteries, which increases the weight of the robot, and therefore, the mobility. It is very important to have a robust platform, and to have all the components easily accessible. The wires should have a good length; not too long, not too short.

12. POSSIBLE IMPROVEMENTS

Imagination has no limits:

- Voice recognition. The noise of the motors could be again an important issue to be considered. The robot would obey the orders of the user.

- To have solar cells to provide the robot solar energy. Anyway, it is difficult to obtain great amounts of power in this way. Also, the PIR's used in this project are indoor, so it would be necessary to change them with another ones.

- The robot could move with legs instead of wheels.

- To incorporate software to recognize the patterns view through the camera.

14. **SOFTWARE**

13.1. Test programs

The next pages contain the test programs that were used in the robot.

13.2. Main structure of the program

Scott, TA for the class, recommends to have this general structure for any ICC program.

```
//These are the includes
#include <hc11.h>
#include <mil.h>
#include <analog.h>


//This must be here
extern void _start(void);

//This is how you declare interrupt service routines
#pragma interrupt_handler TOC2_isr, <OTHER_isr>

//This is for a place holder in the interrupt jump table at the end
#define DUMMY_ENTRY    (void (*)(void))0xFFFF

//Global variable can go here.  Note that any variable that you want
//        to use in an ISR and somewhere in the MAIN should be glo  bal.

//This is an example for setting up the SCI system.
void init_SCI(){
  BAUD |= 0X30;
  SCCR1 = 0X00;
  SCCR2 |= 0X2C;
  SCCR2 &= 0X7f;
}


//This is how to set the RTI rate
/* Initialize RTI to interrupt every 32.768ms */
void init_RTI() {
  PACTL |= 0X03;
  TMSK2 |= 0X40;
}

//Here is main
void main ()
{
        //Call init functions here


}
```

```
//This must go at the END of ALL of your code
//Each entry in this table will correspond to an ISR routine that you need to use.
/**********************************************************

  Initialize interrupts vector jump table

**********************************************************/

#pragma abs_address:0xffd6
void (*interrupt_vectors[])(void) =
        {
        SCI_isr,            /* SCI */
        DUMMY_ENTRY,        /* SPI */
        DUMMY_ENTRY,        /* PAIE */
        DUMMY_ENTRY,        /* PAO */
        DUMMY_ENTRY,        /* TOF */
        TOC5_isr,           /* TOC5 */
        TOC4_isr,           /* TOC4 */
        TOC3_isr,           /* TOC3 */
        TOC2_isr,           /* TOC2 */
        DUMMY_ENTRY,        /* TOC1 */
        DUMMY_ENTRY,        /* TIC3 */
        DUMMY_ENTRY,        /* TIC2 */
        DUMMY_ENTRY,        /* TIC1 */
        RTI_isr,                    /* RTI */
        DUMMY_ENTRY,        /* IRQ */
        DUMMY_ENTRY,        /* XIRQ */
        DUMMY_ENTRY,        /* SWI */
        DUMMY_ENTRY,        /* ILLOP */
        DUMMY_ENTRY,        /* COP */
        DUMMY_ENTRY,        /* CLM */
        _start              /* RESET */
        };
#pragma end_abs_address
```

13.3. Robot's Program

See next pages.