IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

LOBOT - A Review

T.Venu Gopal

* Associate Professor, ECE, Vidya jyothi Institute of Technology,Aziz Nagar,C.B.Post, INDIA

ABSTRACT

Small sized ground robotic vehicles have great potential to be deployed in situations that are either uncomfortable for humans or simply too tedious. For example, a robot may become part of industrial operations, or become part of a senior citizen's life, or become a tour guide for an exhibition center. The robot is kept as small as possible to allow access through narrow passageways such as a tunnel. To fulfill these missions, the robotic vehicle often has to obtain its accurate localization in real time. Considering the difficulty or impossibility in frequent calibration or the management of external facilities, it is desirable to have a self-contained positioning system for the robot: ideally, the localization system should be completely integrated onto the robot instead of requiring external facilities to obtain the position; the system should work indoors and outdoors without any human involvement such as manual calibration or management. Meanwhile, the cost is expected to be as low as possible.

There exist various localization schemes for ground robotic vehicles. These techniques normally utilize GPS, inertial sensors, radio signals, or visual processing. GPS often becomes inoperable in certain environments such as indoors or in wild forests. Additionally, the GPS operations consume power quickly. As an alternative, a localization system may use various waves including electromagnetic waves of various frequencies. The radio-based positioning is among the most popular techniques.

This technology requires a set of external devices to generate or receive radio signal; as the reference nodes, these external devices should have known positions. The accuracy of the radio-based positioning strongly depends on the proper calibration of the reference devices and the target node, as well as a friendly radio environment. Maintaining such a positioning system can be costly and difficult in terms of additional hardware intensive tuning, and environmental management. It is also vulnerable to interference from other signals, thus affecting the accuracy of positioning. Another category of solutions is vision techniques for mobile robot navigation. Generally, these techniques heavily rely on sophisticated techniques on the recognition of an object or shape from images and often have restricted spatial and visional requirements. The performance usually strongly depends on the environment in which the robot operates and the localization suffers frequent failure. Additionally, they may require a known map of the environment. Overall, the vision-based positioning is relatively costly and difficult to implement or maintain.

KEYWORDS: GPS, inertial sensors, radio signals, hardware intensive tuning

INTRODUCTION

Small sized ground robotic vehicles have great potential to be deployed in situations that are either uncomfortable for humans or simply too tedious. For example, a robot may become part of industrial operations, or become part of a senior citizen's life, or become a tour guide for an exhibition center. The robot is kept as small as possible to allow access through narrow passageways such as a tunnel. To fulfill these missions, the robotic vehicle often has to obtain its accurate localization in real time. Considering the difficulty or impossibility in frequent calibration or the management of external facilities, it is desirable to have a self-contained positioning system for the robot: ideally, the localization system should be completely integrated onto the robot instead of requiring external facilities to obtain the position; the system should work indoors and outdoors without any human involvement such as manual calibration or management. Meanwhile, the cost is expected to be as low as possible.

There exist various localization schemes for ground robotic vehicles. These techniques normally utilize GPS, inertial sensors, radio signals, or visual processing. GPS often becomes inoperable in certain environments such as indoors or in wild forests. Additionally, the GPS operations consume power quickly. As an alternative, a localization system may use various waves including electromagnetic waves of various frequencies. The radio-based positioning is among the most popular techniques. This technology requires a set of external devices to generate or receive radio signal; as the reference nodes, these external devices should have known positions. The accuracy of the radio-based positioning strongly depends on the proper calibration of the reference devices and the target node, as well as a friendly radio environment. Maintaining such a positioning system can be costly and difficult in terms of additional hardware intensive tuning, and environmental management. It is also vulnerable to interference from other signals, thus affecting the accuracy of positioning.

Another category of solutions is vision techniques for mobile robot navigation. Generally, these techniques heavily rely on sophisticated techniques on the recognition of an object or shape from images and often have restricted spatial and visional requirements. The performance usually strongly depends on the environment in which the robot operates and the localization suffers frequent failure. Additionally, they may require a known map of the environment. Overall, the vision-based positioning is relatively costly and difficult to implement or maintain.

Additionally, inertial sensors are often used in positioning or navigation systems to detect movement. Different than the radio-based and the vision-based techniques, the operation of inertial sensors is independent of external features in the environment and they do not need an external reference. The inertial sensors mainly comprise accelerometers and gyroscopes (gyros).

An accelerometer measures specific force and a gyroscope measures angular rate. Many inertial systems often require extremely accurate inertial sensors to maintain accuracy, which often causes high cost and calibration difficulty. Being widely available and inexpensive, the accelerometer is often perceived as a solution for localization.

BLOCK DIAGRAM:

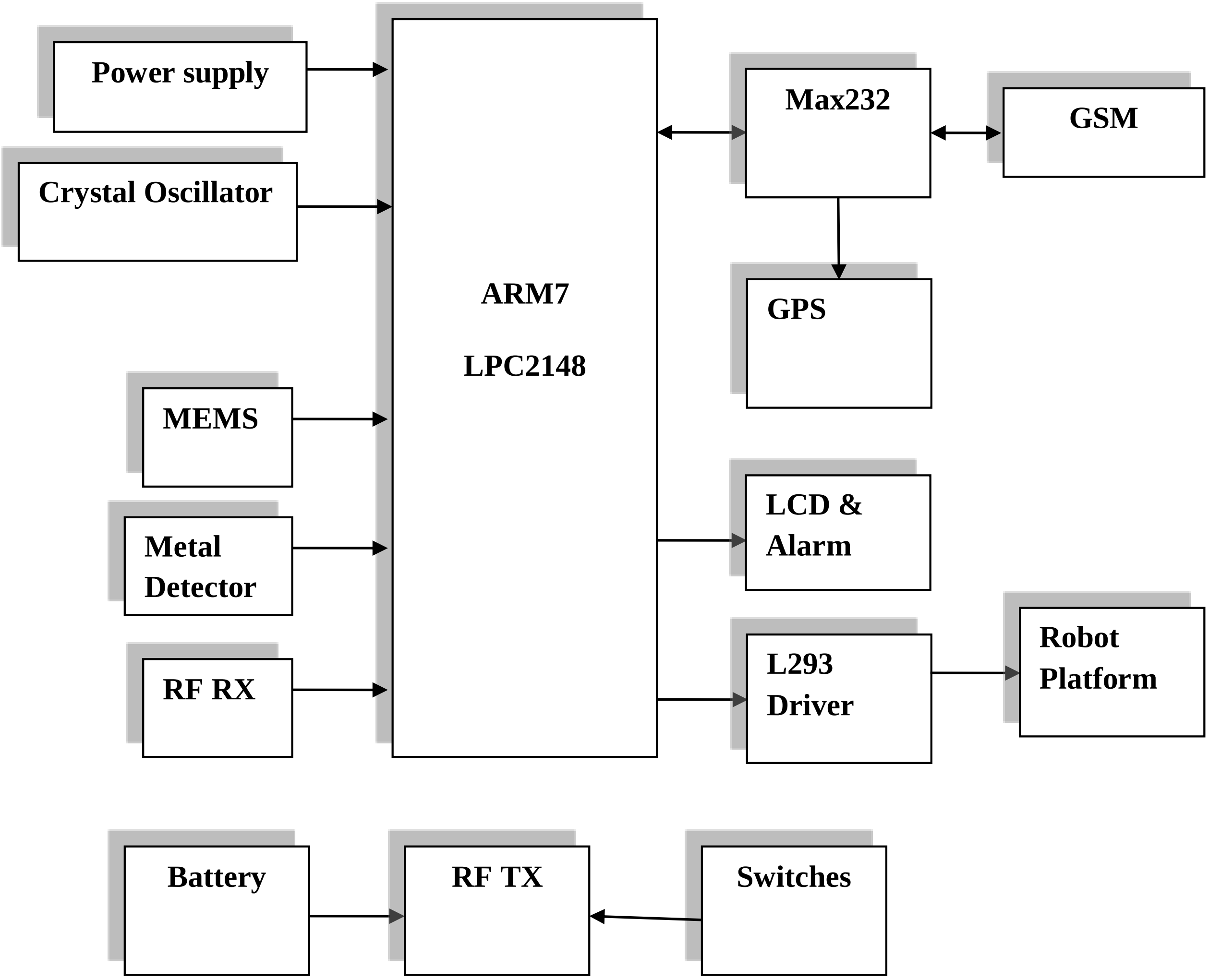


Fig 1.2: Block diagram of LOBOT: low-cost, self-contained localization Of small-sized ground robotic vehicles

DESCRIPTION:

LOBOT localizes a robotic vehicle with a hybrid approach consisting of infrequent absolute positioning through a GPS receiver and local relative positioning based on a 3D accelerometer, a magnetic field sensor, and several motor rotation sensors. All these sensors are installed on the robotic vehicle. The motor rotation sensors are to detect the rotational movement of the motors and thus infer the travel distance of the robot.

An embedded microcontroller inside the robot vehicle takes central control of these sensors and is also responsible for computing the current absolute position. LOBOT infrequently uses GPS to obtain an absolute position and utilizes the accelerometer, the magnetic field sensor and the motor rotation sensors to measure local relative movement since the last known absolute position through GPS. With the GPS data, correction is performed to reduce the cumulative error from the local relative positioning component. The infrequent use of GPS reduces the dependence on the environ-mental impact, e.g., a small area without GPS signal. As a matter of fact, even if GPS is available, LOBOT may still only uses the local relative component over a short time period instead of GPS because GPS is known to have error of up to 20 m while the local relative component has much lower error over a short time elapse. Additionally, the infrequent use of GPS saves electric power.

The local relative positioning component measures the instantaneous 3D moving direction through both the accelerometer and the magnetic field sensor. It also measures the momentary travel distance for every small amount of time elapse through the rotation sensors attached to the vehicle motors. With the moving direction data together with the momentary travel distance, we can obtain an estimate of the movement vector. This seemingly straightforward strategy, however, has encountered a few major technical issues that arise in practical applications. One lies in the distinction between the world reference system and the on-board relative reference system. Another factor that impacts the localization practice is the way the robotic vehicle operates the motors to move. A further complication comes from the cumulative error.

The overall procedure for LOBOT to decide the position.Roughly, the local relative positioning infers the momentary moving orientation and estimates the momentary travel distance with the aid of the accelerometer, the magnetic sensor, and the rotation sensors. The local relative positioning accumulates these momentary estimates to compute the position of the vehicle at any time. Over certain time elapse, the infrequent GPS-augmentation is conducted and is used to perform drift correction so as to obtain better position estimate.

LOBOT is a low-cost, self-contained system. All the necessary hardware devices needed to perform the positioning are a GPS receiver, a 3D accelerometer, a magnetic field sensor, and several motor rotation sensors. LOBOT only needs the commodity versions of these devices that come with moderate precision and low prices. For ease of development, our prototype uses a GPS receiver, a 3D accelerometer, a magnetic field sensor from an unlocked HTC Legend Smartphone that is sold at no more than $300 at the time of this writing. The motor rotation sensors used in this prototype is obtained from a brand of hobby servo motor that sells at $20. Given a complete circuit design, the actual cost of manufacturing a microcontroller chip inte-grating all these raw sensors (including the GPS receiver) can very likely be brought down to well under $100 per set.

Additionally, all these sensing devices including the GPS receiver can be well powered by the battery of the HTC legend Smartphone. Compared with the intense power needed to drive a robotic vehicle, these sensing devices induce only limited overhead in the power consumption. Thus, LOBOT is a low-cost system. The self-contentedness of LOBOT is reflected in two aspects: virtually no requirement of external devices or external facility management; no prior information needed. All the necessary devices are attached to the body of the robotic vehicle that we need to localize. Except for GPS, LOBOT does not require any external devices (e.g., a reference anchor point). The GPS satellite network is maintained by official organizations and thus the use of a GPS receiver virtually needs no effort to maintain external facilities. Unlike many positioning schemes based on vision recognition techniques, LOBOT does not require prior information of the environment either.

METHODOLOGY:

Hardware Requirements:

- MEMS
- LPC2148 CONTROLLER
- GSM
- GPS
- LCD
- L293 DRIVER
- MOTORS

- METAL DETECTOR

Software Requirements:

- KEIL IDE
- EMBEDDED C
- LPC2000 FLASH UTILITY

IMPLEMENTATION OF SOFTWARE

Keil Software

Installing the Keil software on a Windows PC

- Insert the CD-ROM in your computer’s CD drive
- On most computers, the CD will “auto run”, and you will see the Keil installation menu. If the menu does not appear, manually double click on the Setup icon, in the root directory: you will then see the Keil menu.
- On the Keil menu, please select “Install Evaluation Software”. (You will not require a license number to install this software).
- Follow the installation instructions as they appear.

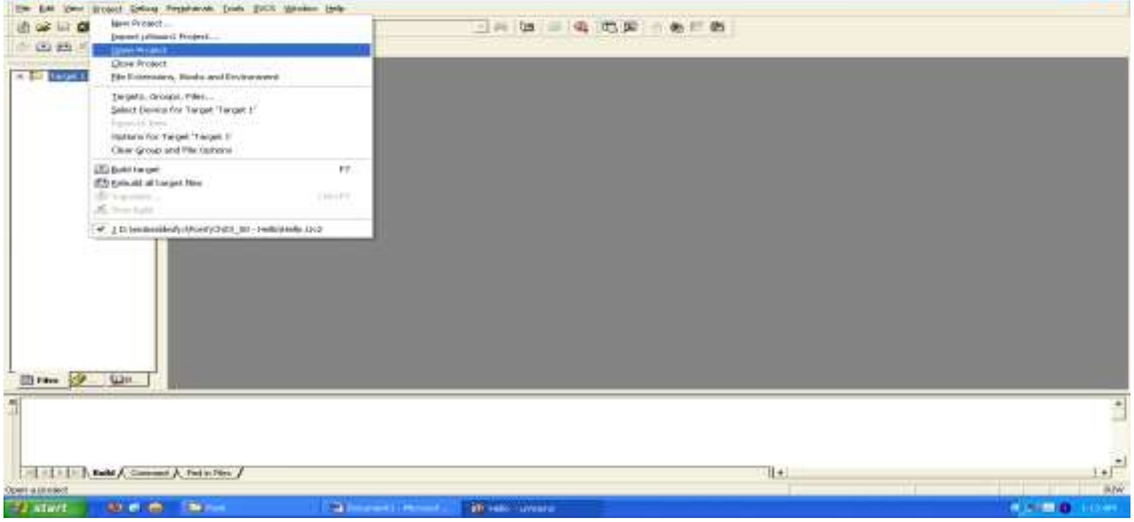
Loading The Projects

The example projects for this book are NOT loaded automatically when you install the Keil compiler.These files are stored on the CD in a directory “/Pont”. The files are arranged by chapter: for example, the project discussed in Chapter 3 is in the directory “/Pont/Ch03_00-Hello”.Rather than using the projects on the CD (where changes cannot be saved), please copy the files from CD onto an appropriate directory on your hard disk

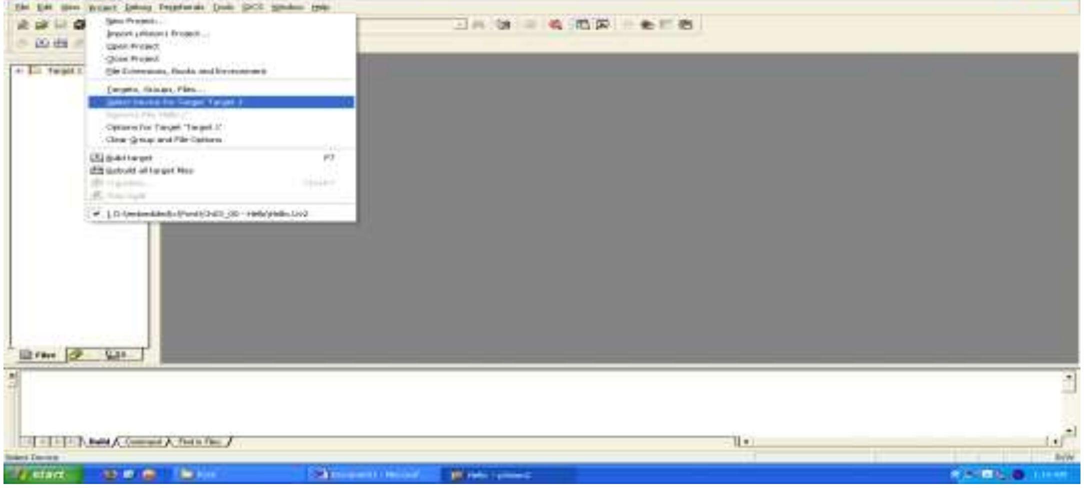
CONFIGURING THE SIMULATOR:

Open the Keil Vision2

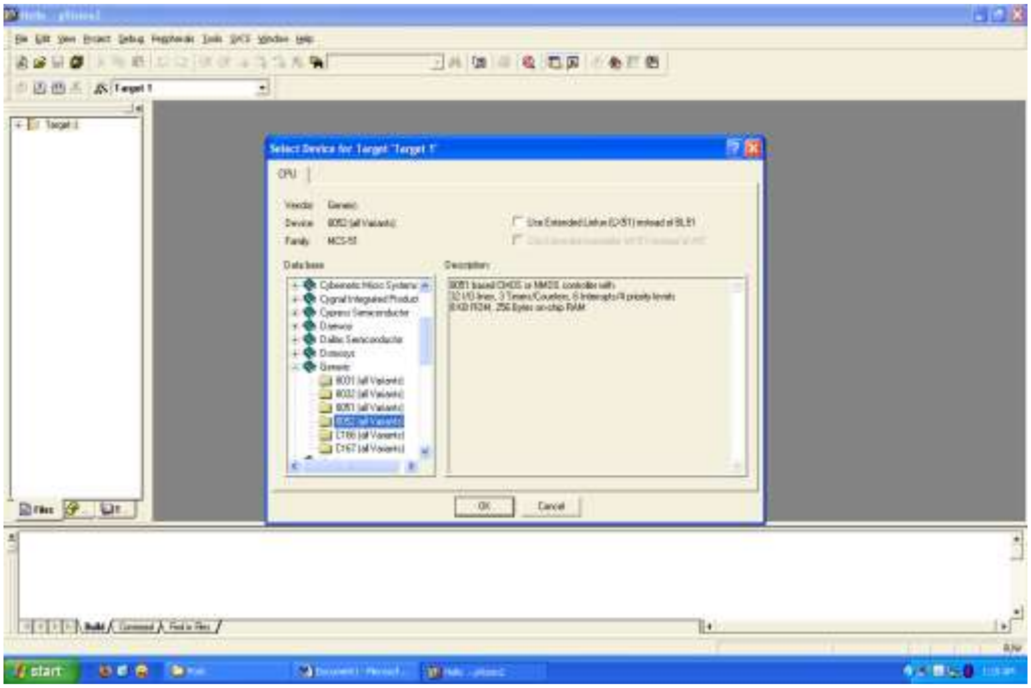
Go to Project – Open Project and browse for Hello in Ch03_00 in Pont and open it.



Go to Project – Select Device for Target ‘Target1’



Select 8052(all variants) and click OK

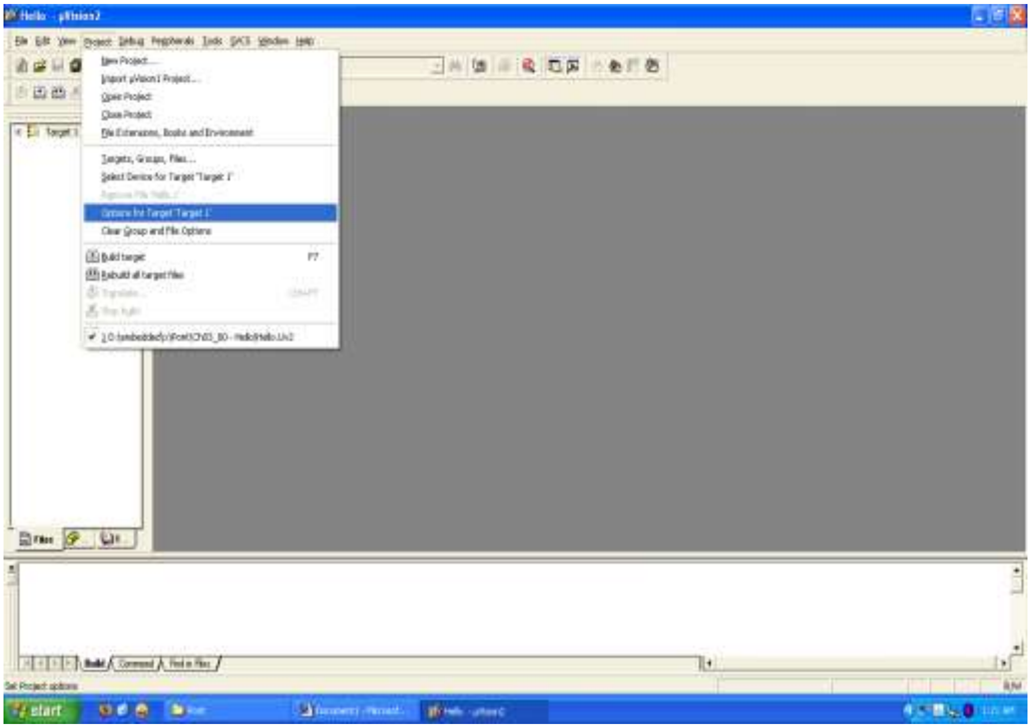


Vendor: Generic
Device: 8052 (all Variants)
Family: MCS-51

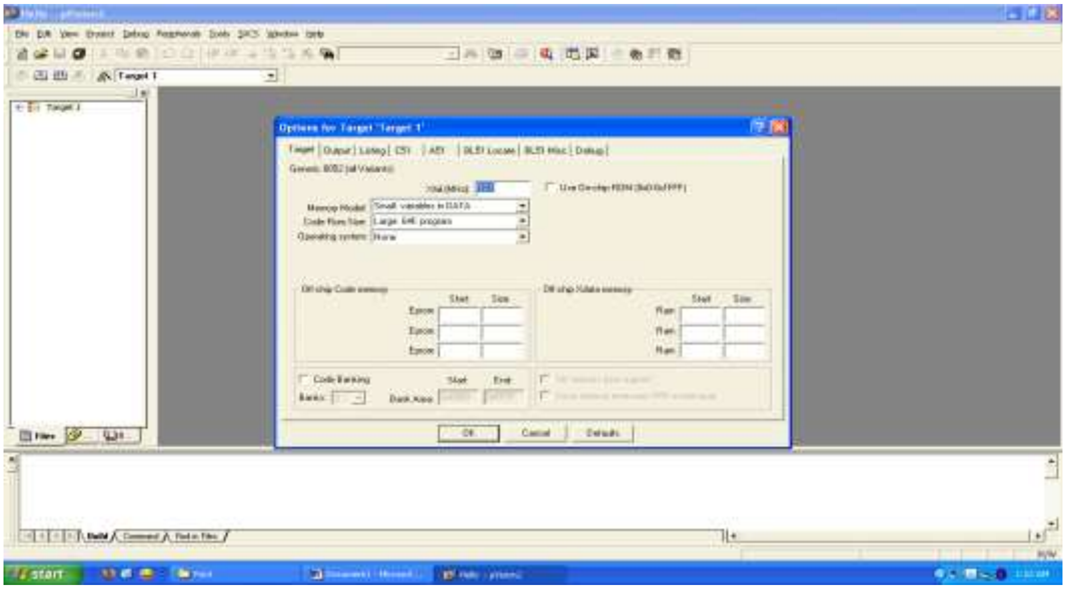
☐ Use Extended Linker (LX51) instead of BL51

8051 based CMOS or NMOS controller with
32 I/O lines, 3 Timers/Counters, 6 Interrupts/4 priority levels
8 KB ROM, 256 Bytes on-chip RAM

Now we need to check the oscillator frequency:
Go to project – Options for Target ‘Target1’

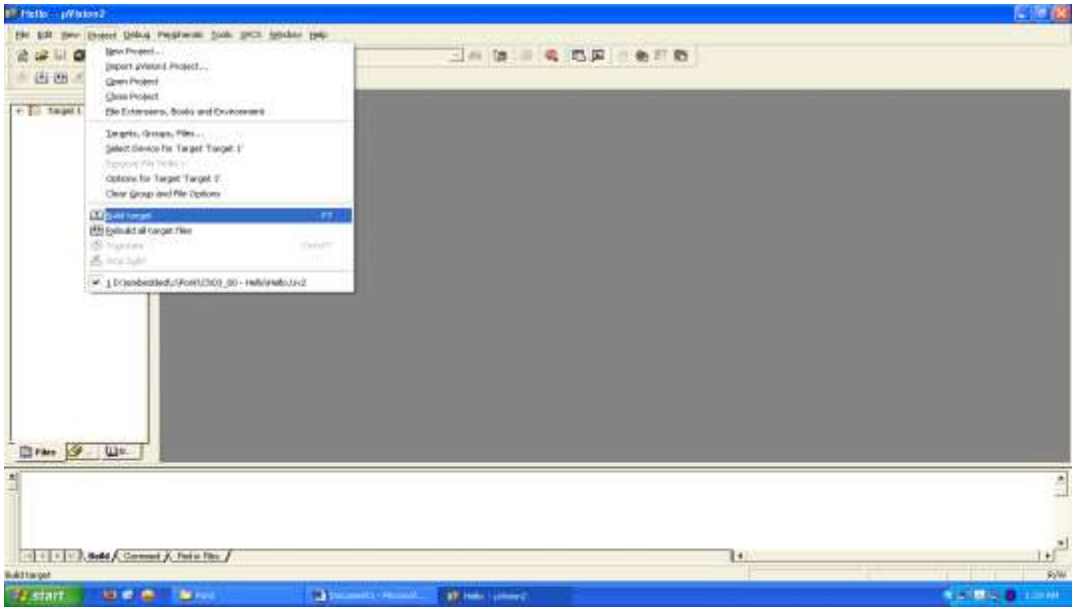


Make sure that the oscillator frequency is 12MHz.



Building the Target

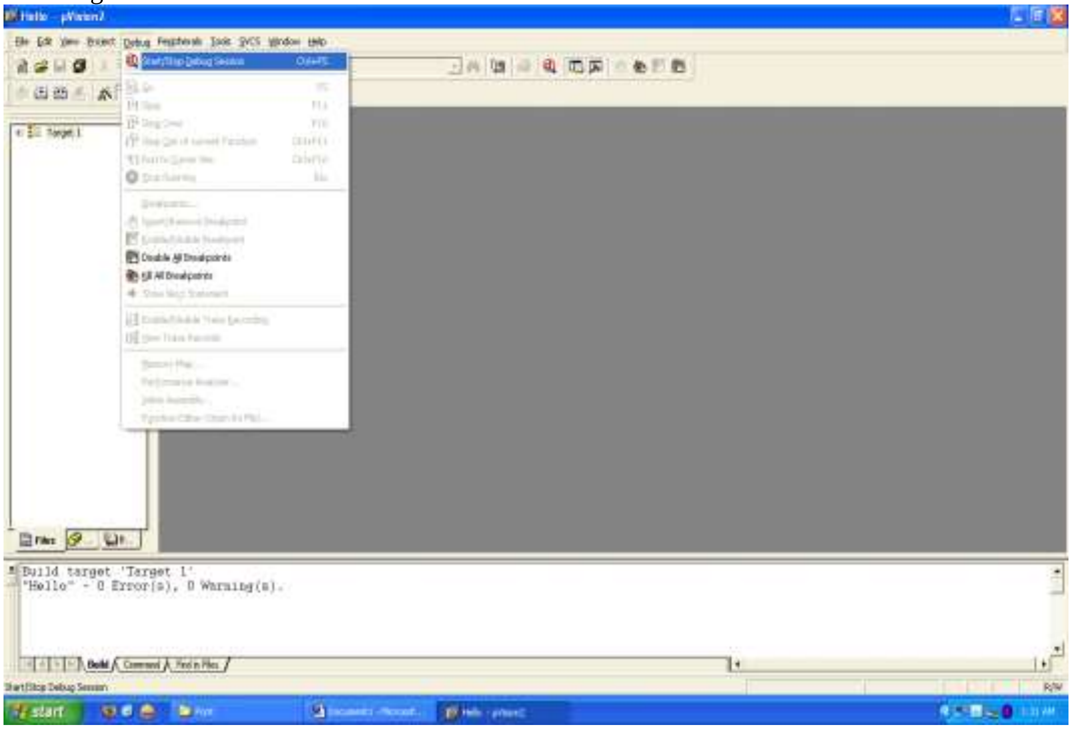
Build the target as illustrated in the figure below



Running the Simulation

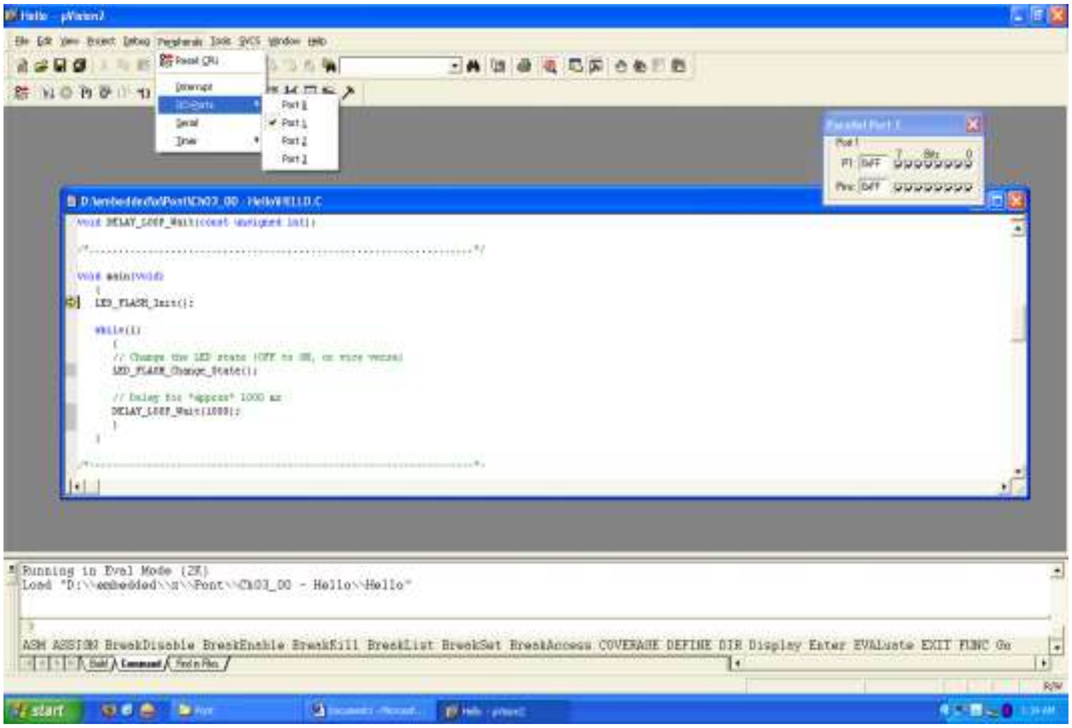
Having successfully built the target, we are now ready to start the debug session and run the simulator.

First start a debug session



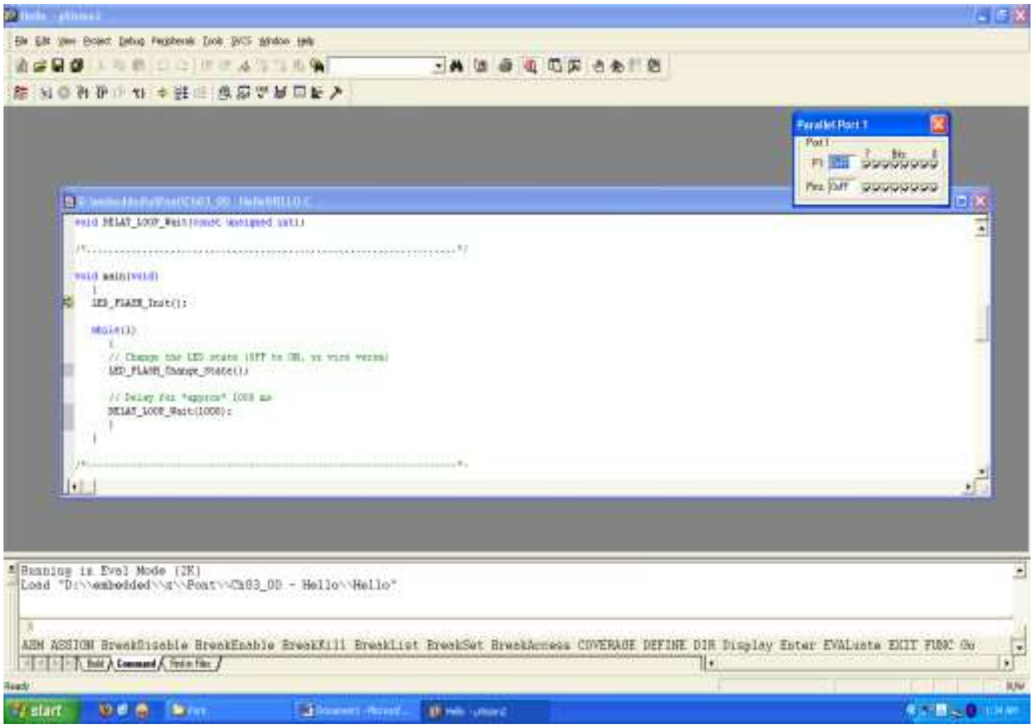
```
Build target 'Target 1'
"Hello" - 0 Error(s), 0 Warning(s).
```

The flashing LED we will view will be connected to Port 1. We therefore want to observe the activity on this port



```
void DELAY_LOOP_Wait(const unsigned int);

/*------------------------------------------------------------------*/

void main(void)
   {
   LED_FLASH_Init();

   while(1)
      {
      // Change the LED state (OFF to ON, or vice versa)
      LED_FLASH_Change_State();

      // Delay for *approx* 1000 ms
      DELAY_LOOP_Wait(1000);
      }
   }

/*------------------------------------------------------------------*-
```

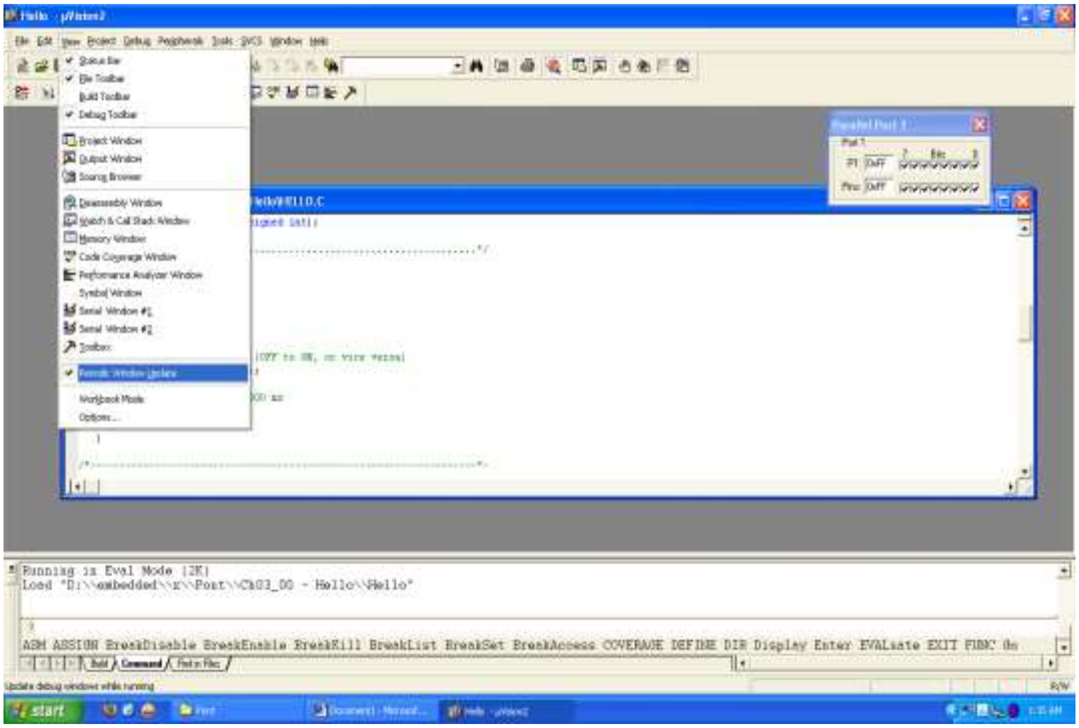
```
Running in Eval Mode (2K)
Load "D:\\embedded\\s\\Pont\\Ch03_00 - Hello\\Hello"

>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```



```
void DELAY_LOOP_Wait(const unsigned int);

/*------------------------------------------------------------------*/

void main(void)
   {
   LED_FLASH_Init();

   while(1)
      {
      // Change the LED state (OFF to ON, or vice versa)
      LED_FLASH_Change_State();

      // Delay for *approx* 1000 ms
      DELAY_LOOP_Wait(1000);
      }
   }

/*------------------------------------------------------------------*-
```

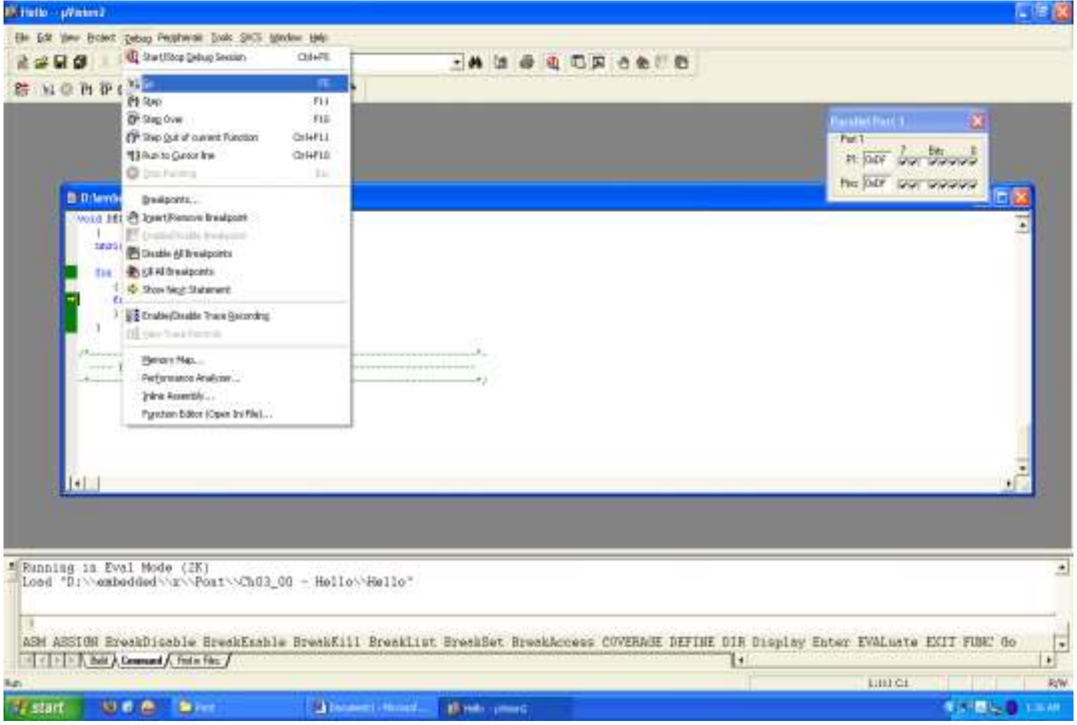
```
Running in Eval Mode (2K)
Load "D:\\embedded\\s\\Pont\\Ch03_00 - Hello\\Hello"

>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```

To ensure that the port activity is visible, we need to start the ‘periodic window update’ flag



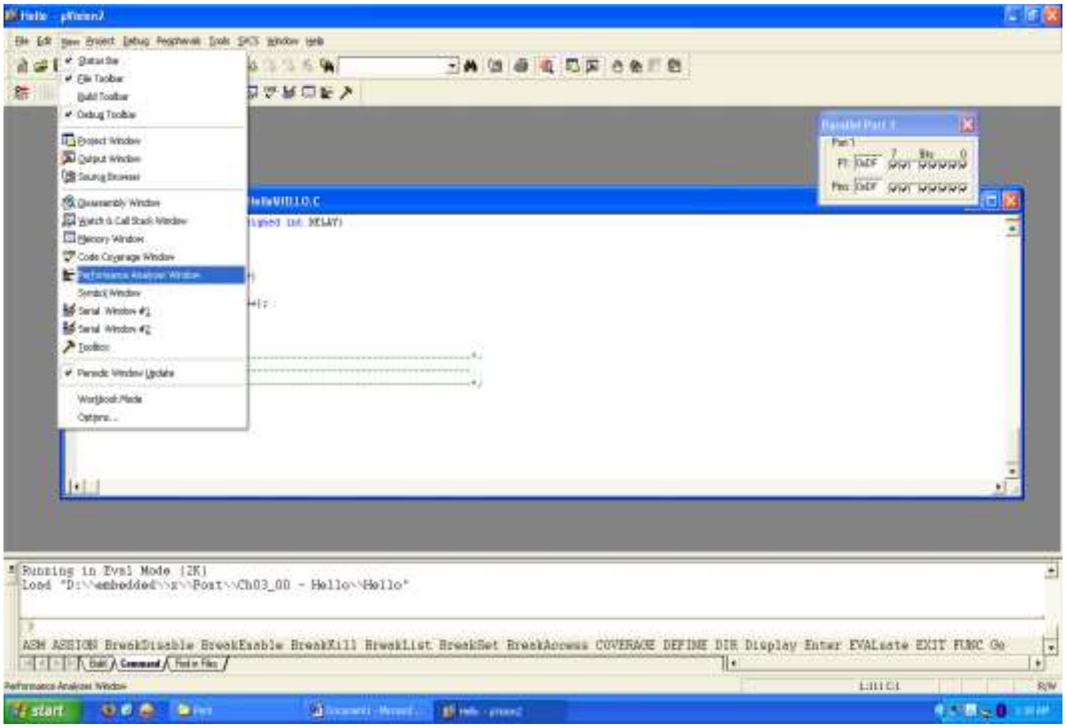
```
Running in Eval Mode (2K)
Load "D:\\embedded\\r\\Port\\Ch03_00 - Hello\\Hello"

>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```

Go to Debug - Go

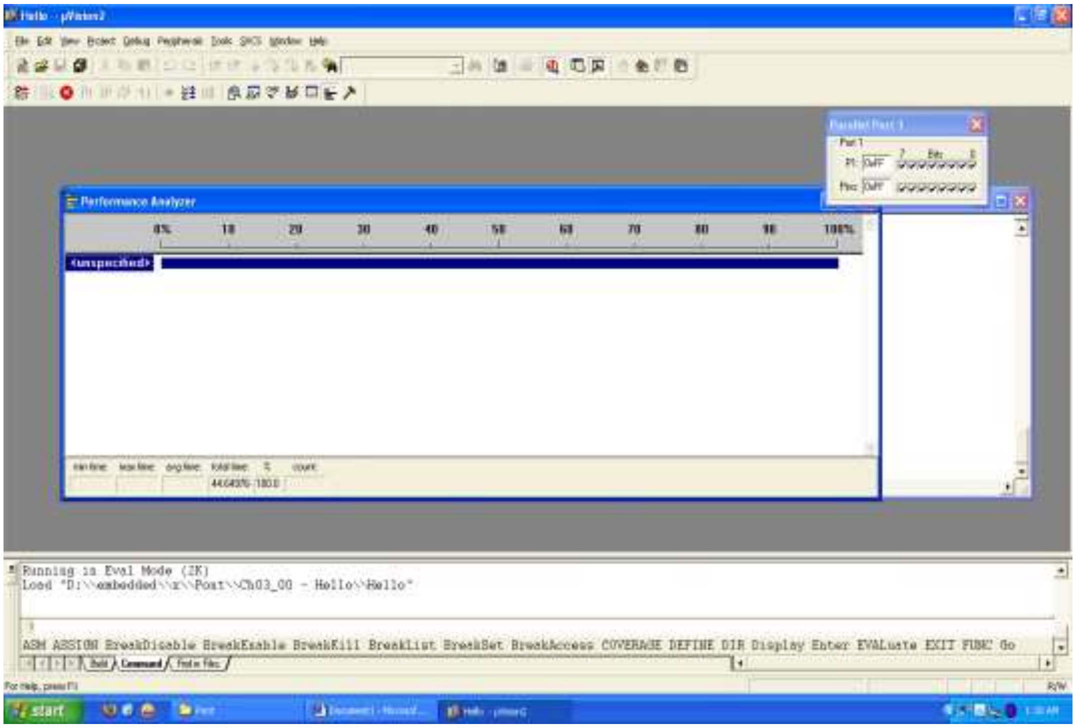


```
Running in Eval Mode (2K)
Load "D:\\embedded\\r\\Port\\Ch03_00 - Hello\\Hello"

>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```

While the simulation is running, view the performance analyzer to check the delay durations.

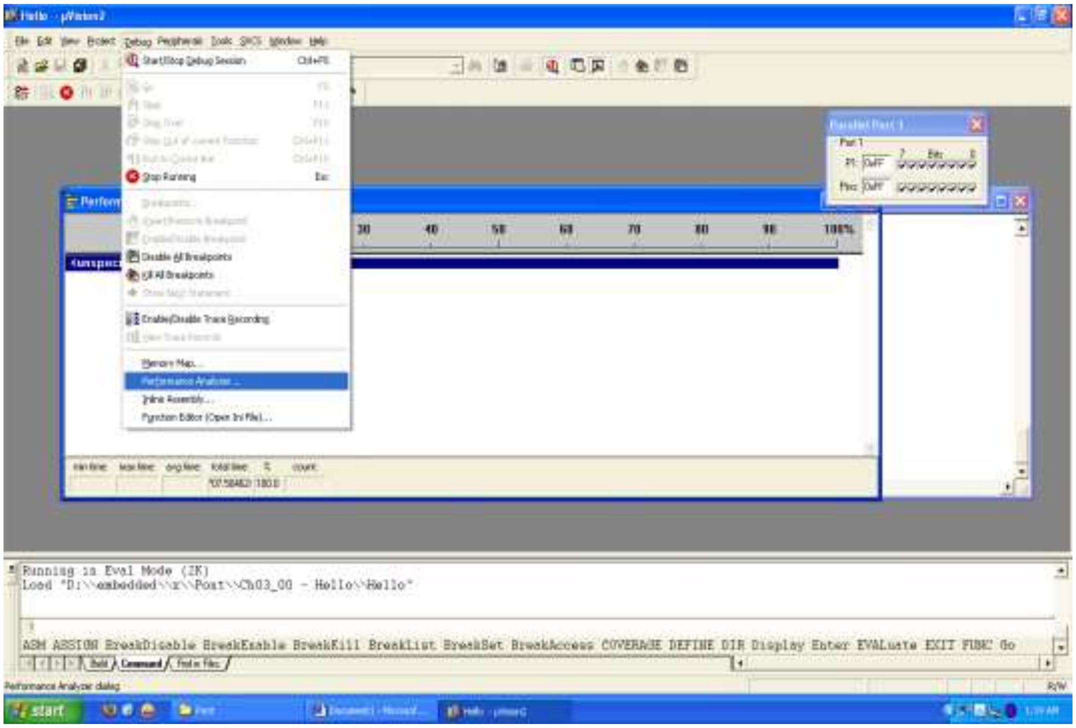


```
Running in Eval Mode (2K)
Load "D:\\embedded\\x\\Port\\Ch03_00 - Hello\\Hello"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```



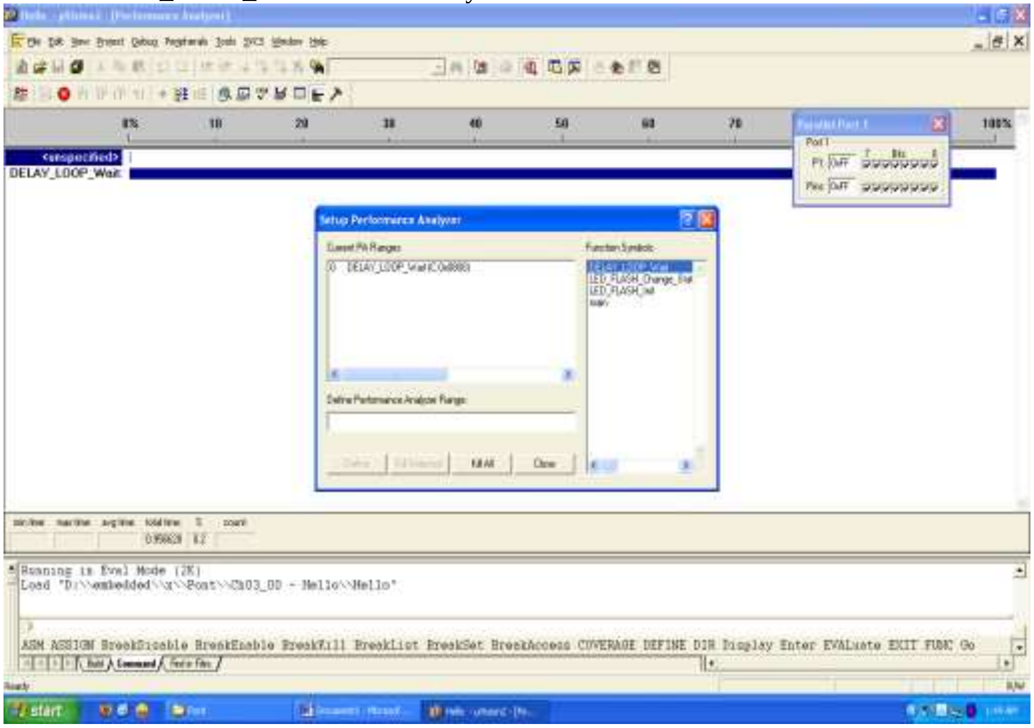
```
Running in Eval Mode (2K)
Load "D:\\embedded\\x\\Port\\Ch03_00 - Hello\\Hello"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```

Go to Debug – Performance Analyzer and click on it



```
Running in Eval Mode (2K)
Load "D:\\embedded\\x\\Port\\Ch03_00 - Hello\\Hello"

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```

Double click on DELAY_LOOP_Wait in Function Symbols: and click Define button



Current PA Ranges:

```
0    DELAY_LOOP_Wait(C:0x8000)
```

Function Symbols:

```
DELAY_LOOP_Wait
LED_FLASH_Change_Stat
LED_FLASH_Init
main
```

```
Running in Eval Mode (2K)
Load "D:\\embedded\\x\\Port\\Ch03_00 - Hello\\Hello"

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
```

Embedded C

Table 2.2 Data types

Data Types	Size in Bits	Data Range/Usage
unsigned char	8-bit	0-255
signed char	8-bit	-128 to +127
unsigned int	16-bit	0 to 65535
signed int	16-bit	-32,768 to +32,767
sbit	1-bit	SFR bit addressable only
Bit	1-bit	RAM bit addressable only
Sfr	8-bit	RAM addresses 80-FFH only

RESULTS AND OPERATION

Figure 1 :LOBOT CIRCUIT

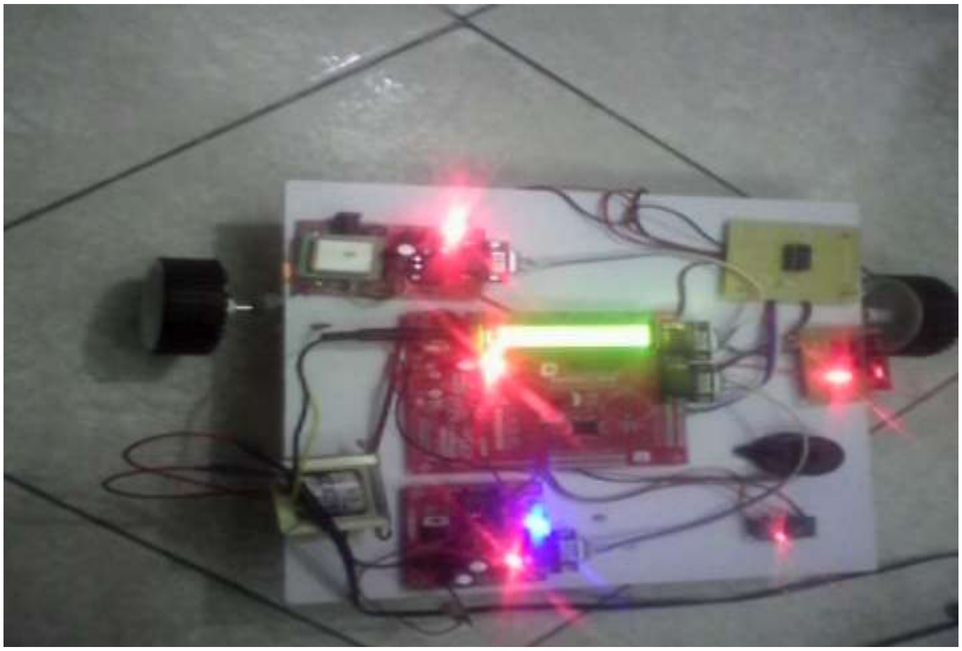


Figure 2: operating GPS and GSM after circuit is ON

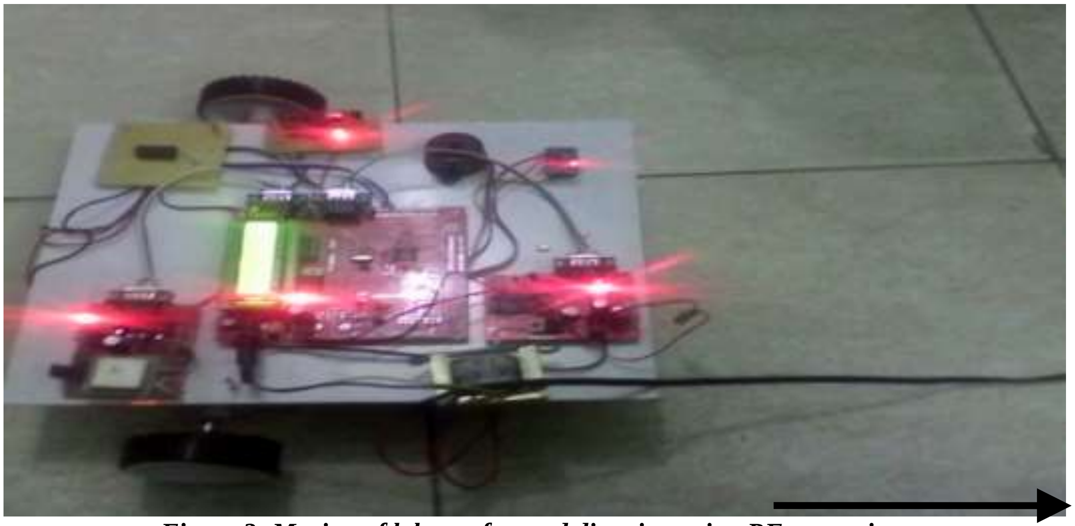


Figure 3: Moving of lobot to forward direction using RF transmitter

CONCLUSION

We propose LOBOT, a low-cost, self-contained, accurate localization system for small-sized ground robotic vehicles. LOBOT localizes a robotic vehicle with a hybrid approach consisting of infrequent absolute positioning through a GPS receiver and local relative positioning based on a 3D accelerometer, a magnetic field sensor and several motor rotation sensors. The hardware devices LO-BOT uses are easily available at low cost. LOBOT is self-contained in that it virtually requires no external devices or external facility management and that it needs no prior information. Unlike other localization schemes such as radio-based solutions, LOBOT does not require external reference facilities, expensive hardware, careful tuning or strict calibration. Additionally, LOBOT applies to both indoor and outdoor environments and realizes satisfactory performance with low cost.

FUTURE SCOPE

This work can be further implemented to design and implement a LOBOT for applications such as fire, medical help, temperature, for cost is expected to be as low as possible. This scheme result in reduce man power, prevent unwanted accidents and alerts user about the incident.

REFERENCES

1. K. Yu and I. Oppermann, "UWB Positioning for Wireless Embedded Networks," Proc. IEEE Radio and Wireless Conf., pp. 459-462, Sept. 2004
2. L. Thiem, B. Riemer, M. Witzke, and T. Luckenbach, "RFID-Based Localization in Heterogeneous Mesh Networks," Proc. Sixth ACM Conf. Embedded Network Sensor Systems (SenSys '08), pp. 415-416, 2008.
3. N. Petrellis, N. Konofaos, and G. Alexiou, "Target Localization Utilizing the Success Rate in Infrared Pattern Recognition," IEEE Sensors J., vol. 6, no. 5, pp. 1355-1364, Oct. 2006.
4. J. Hesch, F. Mirzaei, G. Mariottini, and S. Roumeliotis, "A Laser- Aided Inertial Navigation System (l-ins) For Human Localization in Unknown Indoor Environments," Proc. IEEE Int'l Conf. Robotics and Automation (ICRA), pp. 5376-5382, May 2010.
5. G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp, "Walrus: Wireless Acoustic Location with Room-Level Resolution Using Ultrasound," Proc. Third Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '05), pp. 191-203, 2005. http://doi.acm.org/ 10.1145/1067170.1067191.
6. K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, "The Effects of Ranging Noise on Multihop Localization: An Empirical Study," Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05), pp. 73-80, Apr. 2005.