SPECIFICATIONS FOR MICROMOUSE

For making a micromouse we need to get the specifications about the requirements needed to make it.

1.Sensors
2.Motor and Chasis Design
3.Micro Controller
4.Micro Controller Board
5.schematic Diagrams

Sensors:

We use overhead,down looking, IR sensors for wall detection. The sensor array had 11 individual IR sensors. 4 sensors were present on either wings an 3 were placed at the center (front wing) of the array (refer to the diagrams). The IR sensors are designed to detect obstacle at a specific (1.5cm) distance. The sensitivity of the sensor can be electronically controlled using a potentiometer/preset.

Using IR Diodes and IR photo transistors as transmitters and receivers respectively. To make the IR beams directional we use plastic sheaths (found in CAT5 cables). A parallel plate of plastic should be used to hold the structure tightly.

One lesson we learnt the hard way was that this system is very susceptible to environmental noise. The receivers were unable to distuinguish between the ambient IR in the atmosphere and the IR comming from the diodes. As a result, the sensors worked well in closed environments and in dark but not in direct sunlight.

To overcome this problem, in we should use TSOP1833 instead of IR photo transistors. TSOP1833 detects IR beam oscillating at 33Khz and rejects a constant IR source as noise. To generate the oscillating IR beam we used a 555 timer circuit connected to the IR diode. This setup was much more immune to ambient IR.

One can alternatively use TSOP1738, which detects IR beams oscillating at 38Khz. TSOP1738 is slightly easier to use in our application but it is physically bigger than TSOP1833 and that can be a problem if space is constrained.

Motor and Chasis Design:

Mechanical stability of the robot is very important. One has to make sure that the wheels are perfectly alinged with each other, the weight of components are evenly spread, viabration is minimised etc.

While choosing motors, we can either opt for DC motors or Stepper motors. We should opt for Stepper Motor for high precision and easier contol. A DC Motor gives much better torque to motor weight ratio but is significantly more difficult to control. Among stepper motors you can either go for Uni-polar or Bi-polar. Uni-polar motors are again easier to control but Bi-polar motors provide more torque.

Robot have two wheels directly mounted onto the shaft of the motors. The motors were mounted on the chasis in such a way that their shafts passed through the centre of the robot's body. This symmetric placement of the wheels allow the robot to turn 90 degrees in-place (on a dime). For stability of the robot, two free-sliding castor wheel were placed in the front and rear of the robot.

Micro-Controller:

Philips P89C668 microcontroller for our mouse. It is 8051 compatable microcontroller but has many additional features.

First and foremost, It has **8KB of on-chip RAM**. The first 256 bytes of RAM (including the 127 bytes of indirectly accessible RAM) is identical to what is available in 8051 the rest of the memory is available as external RAM. This means reduced circuitry for the mainboard as external RAM need not be seperately put.

It has **4, 8-bit ports** which can be used for input or output. In a 8051/52 2 ports get used up if external memory is used and only 2 8-bit ports are availabe for input or output. Since external memory is already present on chip, all ports can be used for input or output.

It can operate at high clock frequency and is **twice** as fast as conventional 8051/82 at the same clock frequency.

It supports In-System-Programming(ISP) and In-Application-Programming(IAP). The ISP mode allows us to directly program the microcontroller from any PC (through serial cable)without having to repeatedly remove the microcontroller from the mainboard. The IAP mode allows us to burn data on the flash ROM while the application is executing. This feature is specially helpful in storing partial maze information which can be retrieved even oven a system reset

We can code in sdcc for microcontroller which has similar syntax as c computes 8051 machine code

One can also use Keil, which a commercially available C cross compiler. The compiled code was then transfered to the microcontroller with the help of "flashmagic".

Micro-Controller Board:

The PCB design layout is shown below. The PCB design was made in Eagle software on linux. The PCB is genral purpose in nature and it has 2, 8-bit ports connected to current driver (ULN 2803). These ports can be used as output ports while the other two 8-bit ports can be configured as input ports. The ISP related circuitry(MAX232) is also present on board, and only a few jumpers need to be changed to put the microcontroller in programming mode.