LECTURE 2

ENGR5: Intro to Engineering Practice

MEM Project

LEGO Robo-Soccer

Getting Started

- Start the IC application
- Click on the picture of the RCX
- Click on the “Connect Later” button if you want to create a program
- Click on the “Connect Now” button if you want to download a program already created or to interact with RCX

RCX Firmware

- If your RCX does not say IC 4 when you turn it on, then you will need to download the firmware
- Select “Download Firmware” from the **Settings** menu
- Select appropriate serial port
- Click on “Download Firmware” button
- Follow the onscreen directions

Interacting with IC

- Click on the **Interaction** tab
- Type into the area at the bottom of the IC window
- Examples:

1- Simple Expressions:	2+3;
2- Make some noise:	beep();
3- Check battery status:	battery_volts();
4- Print on the LCD:	printf(“Hello”);

Creating a New Program

- Click on the “New” button (upper left corner)
- Type in the program
- Click on the “Save” button to save the program. Choose a name and directory.

Downloading a Program

- Click on the “Open” button to open your saved program
- Select the tab with your program’s name and click download
- To run your program (**main** function), press the “Run” button on the RCX

Program Structure

```
void main()
{
        /* This is …*/
        statement1;
        statement2;
}

type other_function(type)
{
}
```

- All programs must have a main() function:
  - NO argument is passed
  - NO value is returned
- Instructions to the computer are called statements:
  - Declaration
  - Assignment
  - Mathematical Operation
  - Function Call
- All statements live inside { and }
- All statements are followed by semicolons
- Comments start with /* and end with */
- Indentation makes the program clear
- Program is over when main() is finished

Built-in Functions

MOTORS: void motor(int m, int p)

Turns on motor m at power level p.
Power levels range from 100 for full on forward to –100 to full on backward.

void alloff()

Turns off all motors.

Built-in Functions

TIME: void sleep(float sec)

Waits for an amount of time equal to or slightly greater than sec seconds. sec is a floating point number!!!

```
/*Wait for 5 seconds*/
sleep(5.0)
```

SOUND: void beep()

Turns on motor m at power level p.
Power levels range from 100 for full on forward to –100 to full on backward.

Assignment

Build a program that makes the RCX go forward for 3 seconds, beep, go backward for 3 seconds, stop, and beep.