

Topics for Today

Announcements (Extra credit for any of these)

Thursday Extra: Thursdays@4:00 in Science 3821

Next week: Off-campus Study in CS

eSpeak update

Lab partner preference clarifications

If preference stated,

preference NOT to work with a person always honored

preference to work with a person not honored

Illuminating vignette about my experience learning Russian in high school

started with alphabet, vocabulary, grammar (e.g., verbs of motion)

actual goal for Russian faculty: culture, literature, etc.

Today: Scribbler 2 robots

Capabilities (See course Web “Resources”)

Logistics: batteries, sign out, workstation pairing

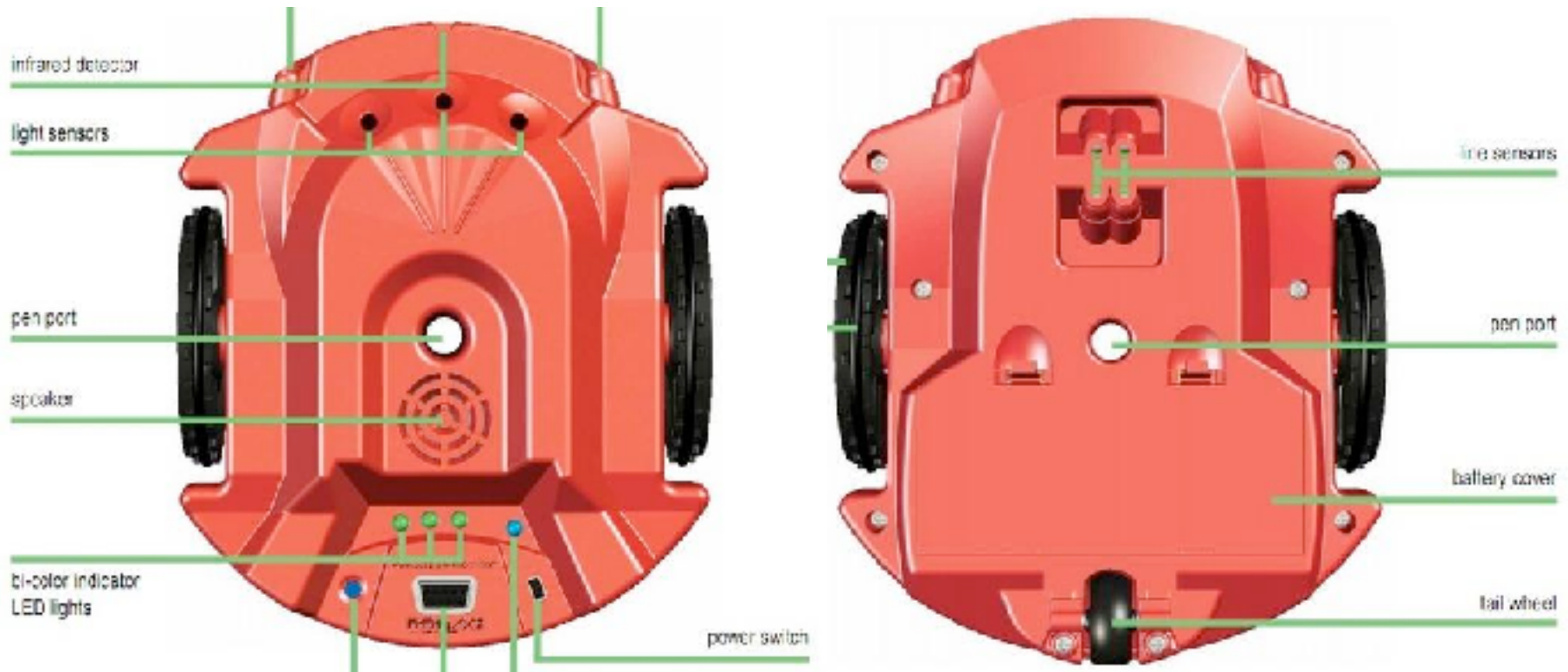
Use (Demo: a sample run)

Programming: First steps

[Simple Input]

Logistics

- Robot ↔ Workstation
- Robots on 4-hour reserve at Science Circulation Desk



Batteries replace (deadBatteries)

/* procedure edited and expanded from version by Prof. Jerod Weinman */

{

if (spaceAvailable(chargers))

 {

 place (deadBatteries, emptyChargerSlots)

 }

else if ((full(chargers) &&

 charged(chargerBatteries))

 {

 place(charged chargerBatteries, greenBasket);

 place(deadBatteries, emptyChargerSlots);

 /* Make sure sensor/light turns on and shows charging */

 }

else

 place(deadBatteries, yellowBasket);

return take(freshBatteries, greenBasket);

}

Getting Started with Robots and MyroC

Program:

- `#include <MyroC.h>`
- `rConnect ("/dev/rfcomm0");` //works for any 3815 workstation
- r-commands: e.g.,
 - `rBeep (2.0, 880);` // beep at 880 frequency for 2 seconds
 - `rForward (0.5, 2.0);` // move forward at half-speed—2 sec.
- `rDisconnect ();`

Compile: requires Makefile

- `make prog`

Run:

- `prog` (or `./prog`, if `.bashrc` file not updated)

Lab time with Robots

Consider the program `input1.c`:

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    int i, j;
```

```
    double x, y;
```

```
    printf ("enter values for i, x, j, y:\n");
```

```
    scanf ("%d", &i);
```

```
    scanf ("%lf", &x);
```

```
    scanf ("%d", &j);
```

```
    scanf ("%lf", &y);
```

```
    printf ("    i    j    x    y\n");
```

```
    printf ("%5d %5d %5.1lf %lf\n",
```

```
           i, j, x, y);
```

```
    return 0;
```

```
}
```

What is printed for input 1 2 3 4?

A.

```
enter values for i, x, j, y:
```

```
  1 2 3 4
```

```
    i      j      x      y
```

```
    1      3     2.0     4.0
```

B.

```
enter values for i, x, j, y:
```

```
  1 2 3 4
```

```
    i      j      x      y
```

```
    1      3     2.000000  4.000000
```

C.

```
enter values for i, x, j, y:
```

```
  1 2 3 4
```

```
    i      j      x      y
```

```
    1      3     2.0  4.000000
```

D.

```
enter values for i, x, j, y:
```

```
  1 2 3 4
```

```
    ???something other than above
```

Consider the program `input1.c`:

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    int i, j;
```

```
    double x, y;
```

```
    printf ("enter values for i, x, j, y:\n");
```

```
    scanf ("%d", &i);
```

```
    scanf ("%lf", &x);
```

```
    scanf ("%d", &j);
```

```
    scanf ("%lf", &y);
```

```
    printf ("    i    j    x    y\n");
```

```
    printf ("%5d %5d %5.1lf %lf\n",
```

```
           i, j, x, y);
```

```
    return 0;
```

```
}
```

What is printed for input

3.1415 2.71828 12 15?

A.

```
enter values for i, x, j, y:
```

```
3.1415 2.71828 12 15
```

```
message about invalid input
```

B.

```
enter values for i, x, j, y:
```

```
3.1415 2.71828 12 15
```

```
    i    j    x    y
```

```
    3    12  2.7 15.00000000
```

C.

```
enter values for i, x, j, y:
```

```
3.1415 2.71828 12 15
```

```
    i    j    x    y
```

```
    3     2  0.1 0.718280
```

D.

```
enter values for i, x, j, y:
```

```
3.1415 2.71828 12 15
```

```
???something other than above
```