

# Sense And Act!

Jeroen de Best

April 20, 2007

Embedded Motion Control (4K450)

## Ready for take-off?

This means (at this time):

- Picked up experimental kit?
- Picked up Embedded Motion Control-CD?
- Cygwin installed?
- BrickOS installed?
- Brushed up C knowledge already?
- Maybe printed "hello world" already?

If not, then hurry up -> count down initiated!

## About the books

What should I read / know?

Extreme Mindstorms, an advanced guide to Lego Mindstorms

- Chapter 7 Introduction to LegOS (BrickOS is known formerly as LegOS)
- Chapter 8 Advanced LegOS
- Appendix A (LegOS part)
- Appendix D LegOS API Reference

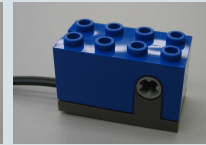
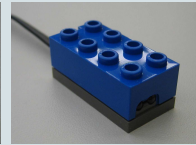
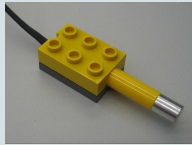
The C Programming Language

- Whole book...

# Sense And Act!

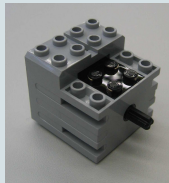
Sensing hardware in your experimental kit

- Temperature sensor
- Light sensor
- Encoder sensor



Acting hardware in your experimental kit

- Motor



## What to do?

Natural question arises:

**How do I control these sensors and motors in BrickOS?**

Explanation for each sensor and motor is done by c-code examples...

## Temperature Sensor (Raw Data)

```
/* include header file: see page 161 Extreme Mindstorms */
#include <dsensor.h>
/* include header file: see page 158 Extreme Mindstorms */
#include <conio.h>
/* include header file: see page 167 Extreme Mindstorms */
#include <unistd.h>

/* begin main: see page 114 C Programming Language */
int main(int argc, char *argv[]) {
    /* begin while loop */
    while(!shutdown_requested()) {
        /* put raw data of sensor port 1 on lcd screen */
        /* see page 158 & 161 Extreme Mindstorms */
        cputw(SENSOR_1);
        /* sleep for 1 second */
        sleep(1);
    }
    /* end while loop */
}
/* return 0 */
return 0;
/* end main */
}
```

What is the temperature in degrees?

# Light Sensor

```
/* include header file: see page 161 Extreme Mindstorms */
#include <dsensor.h>
/* include header file: see page 158 Extreme Mindstorms */
#include <conio.h>
/* include header file: see page 167 Extreme Mindstorms */
#include <unistd.h>

/* begin main: see page 114 C Programming Language */
int main(int argc, char *argv[]) {
    /* Turn on light sensor: see page 163 Extreme Mindstorms */
    ds_active(&SENSOR_1);
    /* begin while loop */
    while(!shutdown_requested()) {
        /* put light value on lcd screen */
        /* see page 158 & 163 Extreme Mindstorms */
        lcd_int(LIGHT_1);
        /* sleep for 1 second */
        sleep(1);
    }
    /* end while loop */
}
/* return 0 */
return 0;
/* end main */
}
```

How is the light on 'Mars'?

# Encoder Sensor

```
/* include header file: see page 161 Extreme Mindstorms */
#include <dsensor.h>
/* include header file: see page 158 Extreme Mindstorms */
#include <conio.h>
/* include header file: see page 167 Extreme Mindstorms */
#include <unistd.h>

/* begin main: see page 114 C Programming Language */
int main(int argc, char *argv[]) {
    /* Turn on encoder sensor: see page 181 Extreme Mindstorms */
    ds_active(&SENSOR_1);
    ds_rotation_on(&SENSOR_1);
    /* Calibrate encoder sensor to zero : see page 181 Extreme Mindstorms */
    ds_rotation_set(&SENSOR_1,0);
    msleep(100);
    /* begin while loop */
    while(!shutdown_requested()) {
        /* put encoder value on lcd screen */
        /* see page 158 & 181 Extreme Mindstorms */
        lcd_int(ROTATION_1);
        /* sleep for 0.1 seconds */
        msleep(100);
    }
    /* end while loop */
    /* return 0 */
    return 0;
}
/* end main */
}
```



# Motor

```
/* include header file: see page 157 Extreme Mindstorms */
#include <dmotor.h>
/* include header file: see page 158 Extreme Mindstorms */
#include <conio.h>
/* include header file: see page 167 Extreme Mindstorms */
#include <unistd.h>

/* begin main: see page 114 C Programming Language */
int main(int argc, char *argv[]) {
    /* define motor direction: see page 156 Extreme Mindstorms*/
    motor_a_dir(fwd);
    /* speed up the motor (0 - 255 = 0 - MAX_SPEED)*/
    motor_a_speed(MAX_SPEED);
    /* sleep for 5 seconds */
    sleep(5);
    /* brake! */
    motor_a_dir(brake);
    /* sleep for 0.1 seconds */
    msleep(100);
    /* turn off motor */
    motor_a_dir(off);
    motor_a_speed(0);
    /* return 0 */
    return 0;
}
/* end main */
```

# One step further to the mission!

We have learned

- a fraction of the c programming language,
- to read out sensors,
- to actuate motors,
- to put some information to the lcd-screen.

More info:

- Embedded Motion Control Course Guide
- Books
- Internet