

```

#include "eyebot.h"
#include <stdio.h>
#define CM_PER_REV (19.7f)
#define TPR_LEFT (40652.0f/40.0f)
#define TPR_RIGHT (43714.0f/40.0f)

ServoHandle servoleft;
ServoHandle servoright;
QuadHandle encLeft, encRight;
struct tcb *drivingthread;
struct tcb *readlightthread;
struct tcb *readkeythread;
struct tcb *readdistancethread;
void driveRaw(float speed, float turn);
void readLight ();
void readKey ();
void readTemp ();
float speedin, turnin;
int key, stop=0, value, PowerON, distance=0, counter=1, counter2=0, turncounter=0, rotate=
4000, KThread=0, status=0, lights;
enum DRIVE_STATE { FORWARD, EVASIVEMANEUVER, TURN, STOP, ONEHUNDREDANDEIGHTY };
enum DRIVE_STATE state = FORWARD;
int evl, evr;
void updateEncoderValues ();

// Driving-tråden: Denne tråden styrer alt med kjøringen kontinuerlig.
void driving()
{
    servoleft = SERVOInit (SERVO12);
    servoright = SERVOInit (SERVO13);

    encLeft = QUADInit (QUAD_LEFT);
    encRight = QUADInit (QUAD_RIGHT);

    float cm = 10.0f;
    int startTicsLeft = 0;
    int startTicsRight = 0;
    int endTicsLeft = 0;
    int endTicsRight = 0;
    float distNow = 0.0f;
    float distRemaining = 0.0f;
    float revDiff = 0;
    float speed = 0.0f;
    float turn = 0.0f;
    float dB, dC, dD, dE;

    do
    {
        switch (state)
        {
            case FORWARD: // Kjører roboten fremmover.
                speedin = 0.2;
                turnin = 0.0;
                break;
            case EVASIVEMANEUVER: // Kjører roboten bakover 10cm, og deretter sender den melding til case TURN: om å svinge
litt til venstre.
                if (cm < 50.0f) {

```

```

    dB = 0.05*cm;
    dC = 0.10*cm;
    dD = 0.90*cm;
    dE = 0.95*cm;
} else {
    dB = 10.0f;
    dC = 20.0f;
    dD = cm - 20.0f;
    dE = cm - 10.0f;
}

updateEncoderValues();
startTicsLeft = evl;
startTicsRight = evr;

endTicsLeft = startTicsLeft + (cm/CM_PER_REV)*TPR_LEFT;
endTicsRight = startTicsRight + (cm/CM_PER_REV)*TPR_RIGHT;

do
{
    updateEncoderValues();
    distNow = CM_PER_REV * ((evl - startTicsLeft)/TPR_LEFT);
    distRemaining = cm + distNow;
    revDiff = ((evr - startTicsRight)/TPR_RIGHT) - ((evl - startTicsLeft)/
TPR_LEFT);

    if(revDiff > 1.0f) {
        turn = 1.0f;
    } else if(revDiff < -1.0f) {
        turn = -1.0f;
    } else {
        turn = revDiff;
    }
    if(distNow < dB) {
        speed = 0.2f;
    } else if(distNow < dC) {
        speed = 0.4f;
    } else if(distNow < dD) {
        speed = 0.6f;
    } else if(distNow < dE) {
        speed = 0.4f;
    } else {
        speed = 0.2f;
    }
    driveRaw(-speed, turn);
}
while(distRemaining > 0.1f);
state=TURN;
break;
case TURN: // Svinger roboten litt til venstre.
    speedin = 0.2;
    turnin = -1.0;
    break;
case STOP: // Stopper roboten.
    speedin = 0.0;
    turnin = 0.0;
    break;

```

```

    case ONEHUNDREDANDEIGHTY: // Snur roboten ca. 180 grader.
        rotate=12400;
        state=EVASIVEMANEUVER;
    default: // Noe er feil, vi skulle ikke vært her.. Derfor stopper vi roboten.
        speedin = 0.0;
        turnin = 0.0;
        break;
    }
    driveRaw(speedin, turnin);
}
while (PowerON==1);

SERVORelease(servoleft | servoright);
QUADRelease(encLeft | encRight);
OSKill(0);
}

void readLight () // ReadLight-tråden: Denne tråden leser signalene fra lys-sensoren kontinuerlig og slår lyset av og på i forhold til avlest
signal.
{
    if (OSGetAD(4)<200) // Her bare skrus lyset på roboten av eller på til å starte med.
    {
        OSWriteOutLatch(0, 0xfe, 0x1);
        lights=1;
    }
    else
    {
        OSWriteOutLatch(0, 0xfe, 0x0);
        lights=0;
    }

    while (PowerON==1) // Her skrus lyset av og på kontinuerlig i forhold til lys-sensoren.
    {
        if (OSGetAD(4)<200 && lights==0)
        {
            AUTone(8000, 50);
            OSWriteOutLatch(0, 0xfe, 0x1);
            lights=1;
        }
        else if (OSGetAD(4)>300 && lights==1)
        {
            AUTone(8000, 50);
            OSWriteOutLatch(0, 0xfe, 0x0);
            lights=0;
        }
    }
    OSWriteOutLatch(0, 0xfe, 0x0);
    OSKill(0);
}

void readKey () // ReadKey-tråden: Denne tråden leser kontinuerlig fra tastaturet. Slik at man alltid kan stoppe roboten eller gjøre andre
ønskelige endringer.
{
    LCDClear();
    LCDSetPrintf(0,0, "Bertha rolling!");
    LCDMenuI(1, (status)? "GO" : "STOP");
    LCDMenuI(2, " EM");
}

```

```

LCDMenuI (3, "180");
LCDMenuI (4, "END");

do
{
    key = KEYGet ();
    switch (key)
    {
        case KEY1: // Sender melding til Driving-tråden om at roboten skal kjøre case: STOP første gang den trykkes og case:
FORWARD den andre gangen og om igjen (stoppe eller starte, mens roboten er i gang).

                if (status==0)
                {
                    state=STOP;
                    status=1;
                }
                else
                {
                    state=FORWARD;
                    status=0;
                }
                break;
        case KEY2: // Sender melding til Driving-tråden om at roboten skal kjøre case: EVASIVEMANEUVER (10 cm tilbake og
deretter litt til venstre, mens roboten er i gang).
                state=EVASIVEMANEUVER;
                break;
        case KEY3: // Sender melding til Driving-tråden om at roboten skal kjøre case: ONEHUNDREDANDEIGHTY (snu roboten
ca. 180 grader mens roboten er i gang).
                state=ONEHUNDREDANDEIGHTY;
                break;
        case KEY4: // Terminerer alle tråder (avslutter programmet) og sender roboten tilbake til startmenyen.
                PowerON=0; // Alle tråder er satt til while(PowerON=1). Så når denne blir satt til 0, blir alle tråder terminert.
                KThread=1;
                LCDClear ();
                LCDPrintf ("Killing Threads.");
                OSWait (50);
                break;
        default:
                break;
    }
    if (KThread==0)
    {
        LCDClear ();
        LCDSetPrintf (0,0,"Bertha rolling!");
        LCDMenuI (1, (status)? "GO" : "STOP");
        LCDMenuI (2, " EM");
        LCDMenuI (3, "180");
        LCDMenuI (4, "END");
    }
}
while (PowerON==1);

AUBeep ();
AUBeep ();
LCDClear ();
LCDPrintf ("Stopped.");
OSWait (50);

```

```

    OSKill(0);
}

void readDistance() // ReadDistance-tråden: Denne tråden leser avstanden til objekter kontinuerlig fra ultralyd-sensoren.
{
    do
    {
        if (OSReadInLatch(0) & 0x40) // Her avleses signalet. Er det 1 (noe som tilsvarer ingen hindringer), settes distance til 0.
        Er det 0 (noe som tilsvarer objekt i veien), settes distance til 1.
        {
            distance=0; // Her settes distance til 0.

            if (state==TURN) // De to if-setningene her, styrer hvor mange grader roboten skal snu.
            {
                turncounter++;
            }

            if (turncounter==rotate)
            {
                turncounter=0;
                rotate=4000;
                state=FORWARD;
            }
        }
        else
        {
            counter++;
            distance++; // Her settes distance til 1.
            LCDSetPrintf(2,0,"C#: %d og D#: %d", counter, distance);
        }

        if (distance > 5 && state==FORWARD) // Vi må ha minst 6 avlesninger om at et objekt er i veien, for å utelukke
        feillesninger.
        {
            counter2++;
            state=EVASIVEMANEUVER;
            AUTone(4000,200);
            distance=0;
        }
    }
    while (PowerON==1);

    OSKill(0);
}

void readTemp() // Metode som leser av temperaturen kontinuerlig og sender ut en tone på 4000Hz i 500ms og sier AUUUUUUUUUUUUU!!!
hvis avlesningene går over 400. Denne skal senere utvides til noe nyttig.
{
    LCDClear();
    LCDPrintf("TESTING TEMP!");

    while (OSGetAD(5) < 400)
    {

    }
    AUTone(4000,500);
    LCDClear();
}

```

```

LCDPrintf("AAAAAAAAAAAAAAAA!!!");
}

void driveRaw(float speed, float turn) // Metode som styrer servoene.
{
    float leftSpeed, rightSpeed;

    leftSpeed = speed + turn;
    rightSpeed = speed - turn;

    leftSpeed = ((leftSpeed > 1.0) ? 1.0 : leftSpeed);
    leftSpeed = ((leftSpeed < -1.0) ? -1.0 : leftSpeed);
    rightSpeed = ((rightSpeed > 1.0) ? 1.0 : rightSpeed);
    rightSpeed = ((rightSpeed < -1.0) ? -1.0 : rightSpeed);

    SERVOSet(servoleft, 128+(int)(127.0f*leftSpeed));
    SERVOSet(servoright, 128+(int)(127.0f*rightSpeed));
}

void updateEncoderValues() // Oppdaterer verdiene for telling av avstand.
{
    evl = QUADRead(encLeft);
    evr = QUADRead(encRight);
}

int main() // Start-menyen.
{
    LCDSetPrintf(1,0,"Welcome to\nBertha.");
    LCDMenuI(1, "GO");
    LCDMenuI(2, "S1");
    LCDMenuI(3, "S2");
    LCDMenuI(4, "END");

    do
    {
        key = KEYGet();
        switch (key)
        {
            case KEY1: // Her opprettes de fire trådene vi har og roboten starter.
                PowerON=1;

                OSMTInit(PREEMPT);

                drivingthread = OSSpawn("driving", driving, 50*1024, MAX_PRI, 1);
                readlightthread = OSSpawn("readLight", readLight, 50*1024, MAX_PRI, 2);
                readkeythread = OSSpawn("readKey", readKey, 50*1024, MAX_PRI, 3);
                readdistancethread = OSSpawn("readDistance", readDistance, 50*1024, MAX_PRI,
4);

                OSReady(drivingthread);
                OSReady(readlightthread);
                OSReady(readkeythread);
                OSReady(readdistancethread);

                OSPermit();
                OSReschedule();
                break;

```

```

case KEY2: // Her kan man lese av signalene fra sensorene. Sensor-menyen del 1.
    LCDClear ();
    LCDSetPrintf (1,0,"Welcome to\nsensors part 1.");
    LCDMenuI (1, "LS");
    LCDMenuI (2, "UZS");
    LCDMenuI (3, "TS");
    LCDMenuI (4, "BACK");

do
{
    key = KEYGet ();
    switch (key)
    {
        case KEY1: // Lys-sensoren-meny.
            LCDClear ();
            LCDSetPrintf (1,0,"Welcome to\nlight-sensor.");
            LCDMenuI (1, "SI");
            LCDMenuI (2, "CON");
            LCDMenuI (4, "BACK");

            do
            {
                key = KEYGet ();
                switch (key)
                {
                    case KEY1: // Her leses signalene fra lys-sensoren for hver gang man trykker.
                        LCDClear ();
                        value = OSGetAD (4);
                        LCDPrintf ("Light-\nsensor: %d", value);
                        OSWait (25);
                        break;
                    case KEY2: // Her leses signalene fra lys-sensoren kontinuerlig til man legger hånda
                        // rundt sensoren og avlesningene går under 20.
                        LCDClear ();
                        while (OSGetAD (4)>20)
                        {
                            value = OSGetAD (4);
                            LCDClear ();
                            LCDPrintf ("Light-\nsensor: %d", value);
                        }
                        break;
                    case KEY4: // Her avsluttes lys-sensor-menyen.
                        stop++;
                        break;
                    default:
                        break;
                }
            }
            LCDClear ();
            LCDSetPrintf (1,0,"Welcome to\nlight-sensor.");
            LCDMenuI (1, "SI");
            LCDMenuI (2, "CON");
            LCDMenuI (4, "BACK");
        }
        while (stop < 1);
        stop=0;
        break;
    }
    case KEY2: // Ultralyd-sensor-menyen.

```

```

LCDClear();
LCDSetPrintf(1,0,"Welcome to\nultrazonic-\nsensor.");
LCDMenuI(1, "SI");
LCDMenuI(2, "CON");
LCDMenuI(4, "BACK");

do
{
    key = KEYGet();
    switch (key)
    {
        case KEY1: // Her leses signalene fra ultralyd-sensoren for hver gang man trykker.
            if (OSReadInLatch(0)&0x40)
            {
                value = 1;
            }
            else
            {
                value = 0;
            }
            LCDClear();
            LCDPrintf("Ultrazonic-\nsensor: %d", value);
            OSWait(25);
            break;
        case KEY2: // Her leses signalene fra ultralyd-sensoren kontinuerlig til man setter
            LCDClear();
            while (counter < 100)
            {
                if (OSReadInLatch(0)&0x40)
                {
                    value = 1;
                }
                else
                {
                    value = 0;
                    counter++;
                }
                LCDClear();
                LCDPrintf("Ultrazonic-\nsensor: %d", value);
            }
            counter=0;
            break;
        case KEY4: // Her avsluttes ultralyd-sensor-menyen.
            stop++;
            break;
        default:
            break;
    }
    LCDClear();
    LCDSetPrintf(1,0,"Welcome to\nultrazonic-\nsensor.");
    LCDMenuI(1, "SI");
    LCDMenuI(2, "CON");
    LCDMenuI(4, "BACK");
}
while (stop < 1);
stop=0;

```

hånda foran sensoren over lengre tid.

```

        break;
    case KEY3: // Temp-sensor-menyen.
        LCDClear();
        LCDSetPrintf(1,0,"Welcome to\ntemp-sensor.");
        LCDMenuI(1, "SI");
        LCDMenuI(2, "CON");
        LCDMenuI(4, "BACK");

    do
    {
        key = KEYGet();
        switch (key)
        {
            case KEY1: // Her leses signalene fra temp-sensoren for hver gang man trykker.
                LCDClear();
                value = OSGetAD(5);
                LCDPrintf("Temp-\nsensor: %d", value);
                OSWait(25);
                break;
            case KEY2: // Her leses signalene fra temp-sensoren kontinuerlig til man får
                // temperaturen til å bli temmelig høy (lighter!).
                LCDClear();
                while (OSGetAD(5)<1022)
                {
                    value = OSGetAD(5);
                    LCDClear();
                    LCDPrintf("Temp-\nsensor: %d", value);
                }
                break;
            case KEY4: // Her avsluttes temp-sensor-menyen.
                stop++;
                break;
            default:
                break;
        }
        LCDClear();
        LCDSetPrintf(1,0,"Welcome to\ntemp-sensor.");
        LCDMenuI(1, "SI");
        LCDMenuI(2, "CON");
        LCDMenuI(4, "BACK");
    }
    while (stop < 1);
    stop=0;
    break;
    case KEY4: // Her avsluttes sensor-menyen del1.
        stop++;
        break;
    default:
        break;
}
LCDClear();
LCDSetPrintf(1,0,"Welcome to\nsensors.");
LCDMenuI(1, "LS");
LCDMenuI(2, "UZS");
LCDMenuI(3, "TS");
LCDMenuI(4, "BACK");
}

```

```
    while (stop < 1);
    stop=0;
    break;
case KEY3: // Her skal det legges inn sensor-meny del 2. Hvor blant annet akselerometeret vil inngå.

    break;
case KEY4: // Her avsluttes start-menyen.
    LCDClear();

    if (stop < 1)
    {
        LCDPrintf("ONE MORE TIME!");
        OSWait(25);
    }
    stop++;
    break;
default:
    break;
}
LCDClear();
LCDSetPrintf(1,0,"Welcome to\nBertha.");
LCDMenuI(1, "GO");
LCDMenuI(2, "S1");
LCDMenuI(3, "S2");
LCDMenuI(4, "END");
}
while (stop < 2);

LCDClear();
LCDPrintf("Terminated\n");

return 0;
}
```