

Lab 7

Distance Sensors:

PSD sensors location:

Left	Centre	Right
PSD 1	PSD 2	PSD 3

Reading Encoder:

PSDGetRaw([psd#])

Servos:

Servo location:

Camera
Servo 1

SERVOSet(int servo, int angle);

Set servo [1..14] position to [0..255]

SERVOSet(1, 128); → Set the servo to the middle

Driving the car:

VWStraight(int dist, int lin_speed);

Drive straight, dist [mm], lin. speed [mm/s]

VWStraight(100, 100); → 10cm @ 10cm/s

VWTurn(int angle, int ang_speed);

Turn on spot, angle [rad/100], ang. speed [(rad/100)/s]

VWTurn(314,600); → 180° @ 1 rev/sec approx.

VWCurve(int dist, int angle, int lin_speed);

Drive Curve, dist [mm], angle [rad/100], lin. speed [mm/s]

*VWCurve(100,10/10*100, 100) → Drive 10cm of a curve of 1 rad angle @ 10cm/s*

- Hint: These functions overwrite any other previous VW function when called.

Example:

Driving a car straight in an infinite loop. The car will drive straight @ 1 cm/s. The function VWStraight overwrites the previous VWStraight.

```
int main()
{
  while (1)
  {
    VWStraight(100, 100);
  }
  return 0;
}
```

VWDriveDone()

Can be used as a flag in order to know if the car has completed a VW task or not.

- Hint: VWDriveDone returns 1 when VW is not used. Therefore in order to use this function properly it is better first to create a loop waiting for VWDriveDone to be 0 and then another loop waiting to be 1.

```
VWCurve(100, 100,100);
while (VWDriveDone())
{
    printf("waiting to be cleared\n");
}
while (!(VWDriveDone()==1))
{
    printf("waiting to be set\n");
}
printf("finished\n");
```

Key Functions:

There are 4 keys in the interface:



The values KEY1, KEY2, KEY3, KEY4 are defined

int KEYGet(void)

Blocking read (and wait) for key press (returns KEY1..KEY4)

int LCDMenu(char *st1, char *st2, char *st3, char *st4)

Set menu entries for soft buttons

int LCDSetPos(int row, int column)

Set cursor position in pixels for subsequent printf

int LCDPrintf(const char *format, ...)

Print string and arguments on LCD

Example:

```
int main()
{
    LCDMenu(" ", " ", " ", "End");
    while (1)
    {
        KEYCODE x = KEYGet();
        if (x==KEY4) break; //END
        LCDSetPos(10,10);
        LCDPrintf("Key: %d\n", x);
    }
    return 0;
}
```

Camera:

```
int main()
{
    BYTE img[QQVGA_SIZE]; // 160x120

    LCDMenu(" ", "", "", "End");
    CAMInit(QQVGA);

    while (1)
    {
        CAMGet(img);
        LCDImage(img);
    }
    CAMRelease();
    return 0;
}
```

Getting an image, displaying it and drawing a circle:

```
2  #include "eyebot.h"
3
4  int main()
5  {
6      BYTE img[QQVGA_SIZE]; // 160x120
7
8      LCDMenu(" ", "", "", "End");
9      CAMInit(QQVGA);
10
11     while (1)
12     {
13         CAMGet(img);
14         LCDImage(img);
15         LCDCircle(160/2, 120/2, 10, ORANGE, 0);
16         usleep(33333);
17     }
18     CAMRelease();
19     return 0;
20 }
```



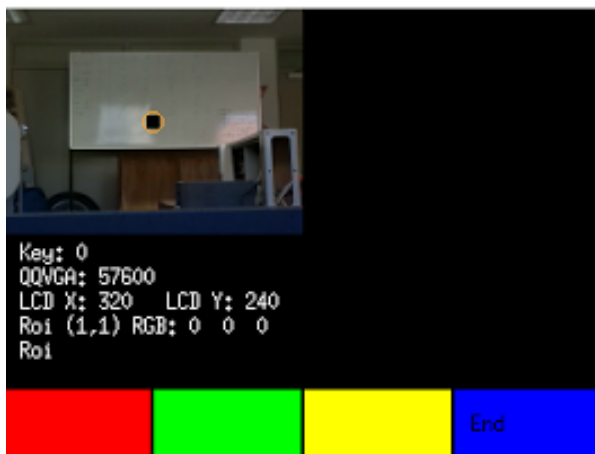
Image Processing:

Converting an RGB image to HIS:

```
1 // EyeBot Demo Program: Camera Display, T. Bräunl, June 2015
2 #include "eyebot.h"
3 #define X_1 160
4 #define Y_1 120
5
6 int main()
7 { BYTE img[QQVGA_SIZE]; // 160x120
8
9     LCDMenu(" ", "", "", "End");
10    CAMInit(QQVGA);
11
12    //while (KEYRead() != KEY4)
13    while (1)
14    {
15        float Hue[X_1*Y_1], Sat[X_1*Y_1], Int[X_1*Y_1];
16        CAMGet(img);
17        LCDImage(img);
18
19        //IPCol2HSI      int IPCol2HSI(BYTE* img, float* h, float* s, float* i)
20        // h,s,i matrix
21        IPCol2HSI(img, Hue, Sat, Int);
22        printf("Hue: %f\n", Hue[(120/2)*X_1+160/2]);
23    }
24    CAMRelease ();
25    return 0;
26 }
27
28
```

Accessing to pixels in an image:

```
//draw a box 6 by 6 pixels in the middle of the image
for (int i=-3;i<4;i++)
    for (int j=-3;j<4;j++)
        for (int rgb=0;rgb<3;rgb++)
            img[((120/2+i)*X_1+(160/2+j))*3+rgb]=0;
```



Using Key functions for getting a new frame or exiting:

```
1 // EyeBot Demo Program: Camera Display, T. Bräunli, June 2015
2 #include "eyebot.h"
3 #define X_1 160
4 #define Y_1 120
5
6 int main()
7 { BYTE img[QQVGA_SIZE]; // 160x120
8
9   LCDMenu("Cont. ", "", "", "End");
10  CAMInit(QQVGA);
11
12  //while (KEYRead() != KEY4)
13  while (1)
14  {
15
16    KEYCODE key= KEYGet();
17    if (key==KEY4) break; //END
18
19    CAMGet(img);
20    LCDImage(img);
21
22  }
23
24  CAMRelease ();
25  return 0;
26 }
27
```