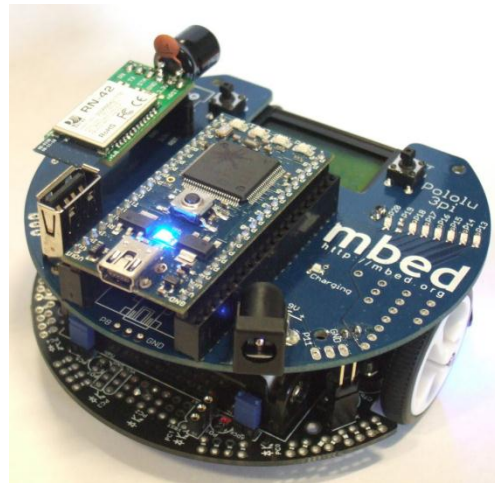
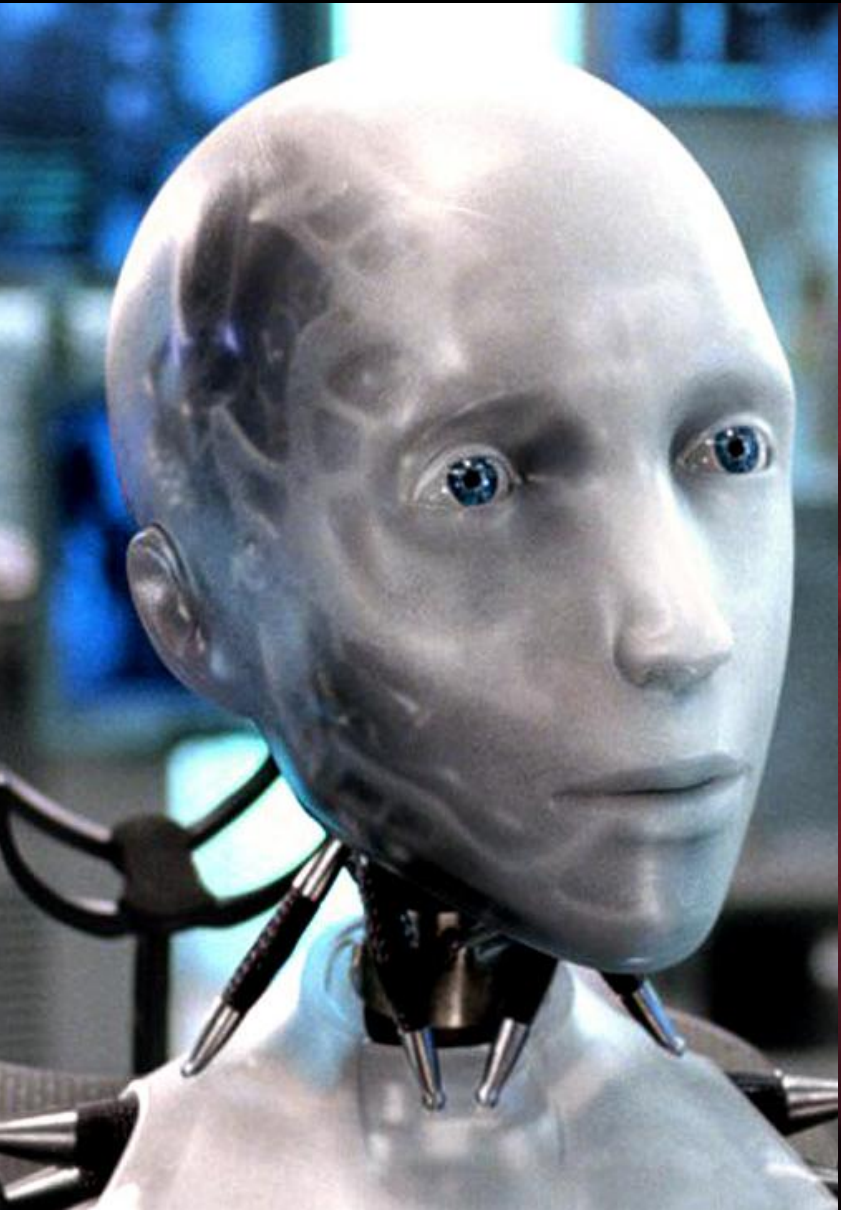


# Robot workshop



# Aims of this workshop

- You should have already used mbeds this morning to program traffic lights.
- We are going to be interfacing the mbed with more complicated hardware
- Somebody else has done a lot of the hard work for you. This means you don't need to build the hardware (no breadboard) and you don't need to do low level control.



# API

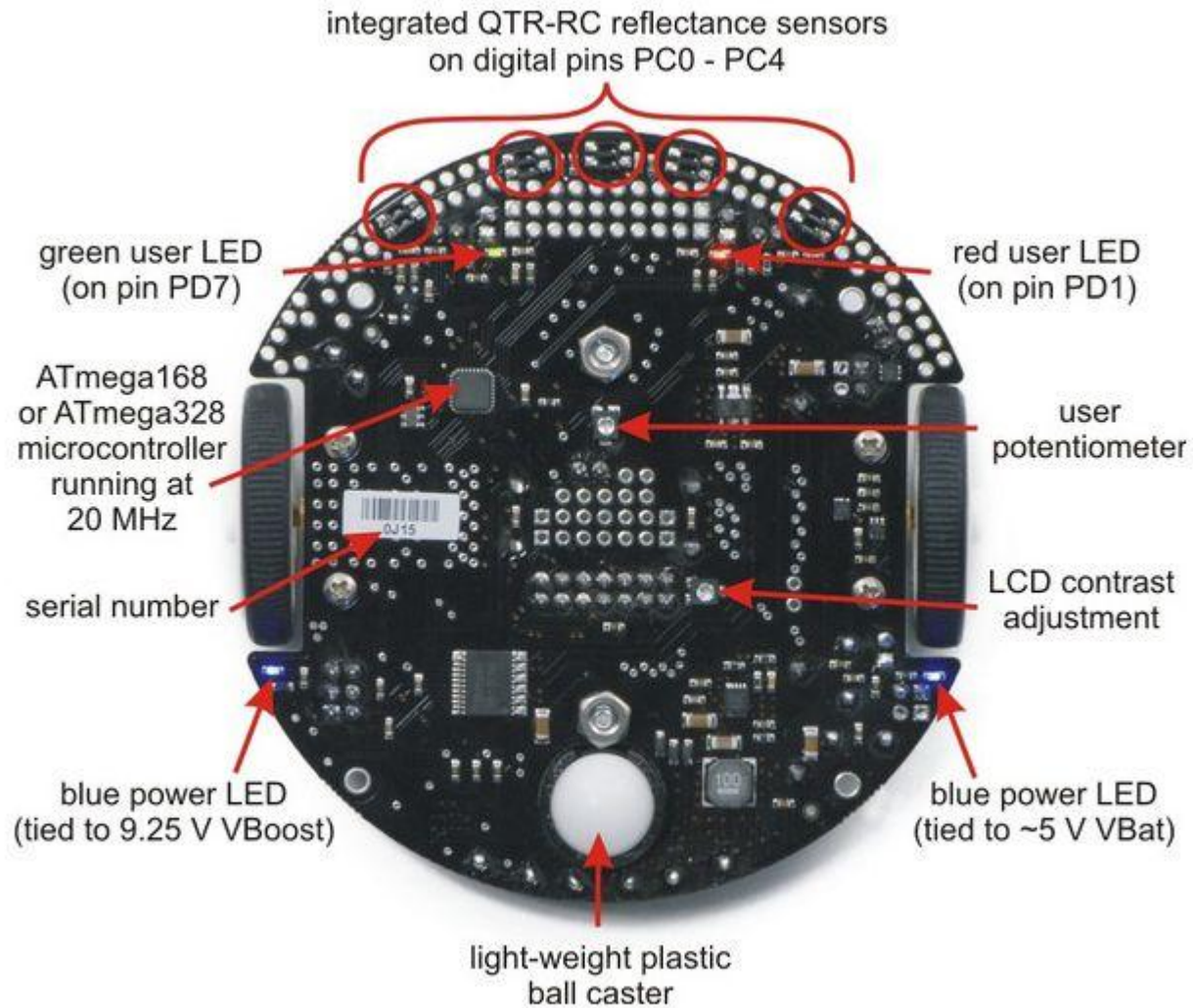
- API stands for application programming interface.
- It is used when you want to write a program for a particular system
- An API makes it much easier to program for particular hardware
- We will use the m3pi API, which includes many functions (already written) that will make programming easier

# Why use API?

- If the functions are already written, you can spend less time writing boring things and focus on more exciting things.
- Our programming now is going to be more “high level”, using functions that will control hardware rather than single commands. This means we can write more complex and more intelligent programs.

# First look at the robot

- This is what your program will interface with, so it's useful to have a basic knowledge of how it works.
- Look at the motors. These are individually controlled, and operating these at different speeds will cause the robot to turn.
- Look for five small black square around the front of the robot. These are the light sensors that will tell it where the line is.
- Look at the LCD screen and LEDs. These form different output methods, which can also be controlled by the mbed.
- The API will do all of these things automatically with functions.



# Line follower

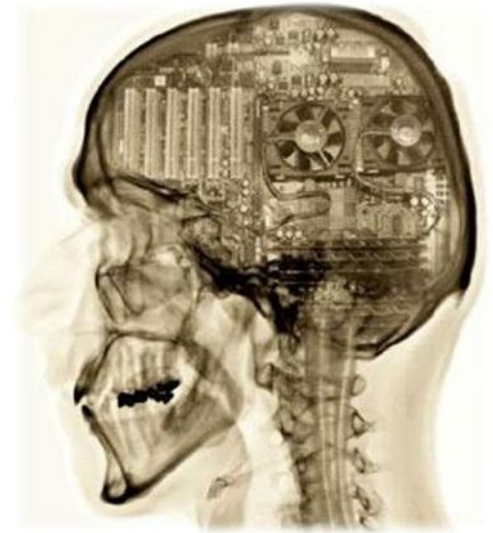
- The task is to make the robot follow a line.
- Imagine telling a human to follow a line, no problem. Even Jim the frozen cowboy could manage it...
- (probably)





# Why is this an intelligent program?

- The robot (mbed) will need to make **decisions** based on outside stimuli. In this case the amount of light the sensors can see.
- Writing an intelligent program is particularly rewarding, because it does appear to be intelligent.
- Intelligent programs are used in advertising, for example find an advert on google. It will be linked to searches that you have made in the past.



# How to proceed

- First of all, the constants used to make the decisions may not be that good.
- Changing these could make it faster.
- Secondly, it could work really well when the robot is in some places on the line, and not so well in others.
- Adding more conditions for different parts could make it perform better in those circumstances.

# Without further ado

- Go to your computers and open the compiler.
- Follow the instructions in your documents, and attempt to get the hello world program working
- Once you have done this, show a demonstrator and they can give you the line following task.
- Make your robot follow the line as fast as possible.
- There may be a prize for the fastest...
- More importantly though
- Have fun!

# HINTS

- Try to make your code readable, this makes it easier to debug, as shown here.

```
it
(r,i,
a,b,c,u,
u,v,m)
double
r,i,a,
k,u,
.
u, v:(double(h), c
= r, d=i,x,u;int(j)=0 {1
=if p=sqrt(1);while(1 >0)
{t++;x=r+r;y=i+i;if(x+y>u)
}break;h=r;r=x-y;a=i-2*k*h
i+h;if(fabs(r-c)<u&&fabs(i-d
)(u)&f1=0;break;}if(j>p){j=0;c
=r,d=i;}1--;return+1;}main(u
,z)char**z;{FILE*fd;double(a), b
u,r,i,x,t,u,v,h,col(64);long(xm)=
u?atoi(x+z+5):300,ym=(a(u)?atoi
(6)):300,x,u;int(c)=0,j=0,k,l,m=(u
(z+8))-100,n,o,p;char(lx)=0>>8,hoc
=ym>8;hy=ym%256;n=(u>9)?(z[9][0]
(u>9)?(s'-(z[9][1][32]+1)):0;co
o;h=clock();do{col=l-(u)+1)?atof
(x+z+2):-2;}while(++c(4);f=fopen("
(u)=11)?*(z+10);n? "Mandelbrot" : "u"
);if(!f)return+1;fprintf(f,"%8a%85%8i%88%8o%8o
");fputc(0,hx-1,lx,hy-1,l,y,hx,lx,hy,l,y);for(l=0;l<
48;l++)fputc(l<<2,f);fprintf(f,"%c%c%c%c%c%c%c%c\n",0,hx
+1)--co/xm;u=(m(co+3)-m(co+2))/c
(ym-1,l,u++)<for(x=0;x<xm-1;x++)<
(co+3)-ym; a=(0)?s:(u>11)?atof(z
(n)?1:(u>12)?atof(w(z+12)):10);
i=(n)?t:0;i=it(r,i,a,b,e,u,v,m)
)?k=it(r,n,i,-v,a,h-u,v,m)
n);p=++(j&&1);i:(j&&1);i:(t&&1)
i:(k&&1);(p=1);j=1;fputc(0?
70:63);<<(t)?70:(63-p%63);f);
printf(" %3.2f%%", (u+1)/ym);
100;0);fflush(stdout);j=(u)
13)?(w(131132))-':0;fputc
(12);b=lock(1);FILE*fd;
a;M?<n>?b?h?EL?i?C?;
for(k=0;k<767;k++){I
767 -k;k?fputc((1/3)*w(
4);f);fclose(f);
puts
(COPY)
puts
USAGE);
return(
0));
}
```

# HINTS

- ASK for help if you need it.
- Struggling by yourself is the worst way to learn!

```
printf("Help I'm stuck!\n");
```

# HINTS

- Programming can seem difficult, if you can't understand what is happening, go through it statement by statement until you understand.
- It is completely logical so it is usually possible to work out what the problem is.



**DON'T  
PANIC**

# What's next?

- First, you need to get the robot working, and make sure that all of the connections to the mbed are correct.
- Instructions on how to do this first part, The “Hello world” program, are in the worksheets.
- Go to <https://mbed.org/cookbook/ARM-Summer-Students-2013-Workshop> to download these and the code.
- If you are unsure, ask a demonstrator, it's what we are here for 😊
- Once this works, you need to write your first intelligent program