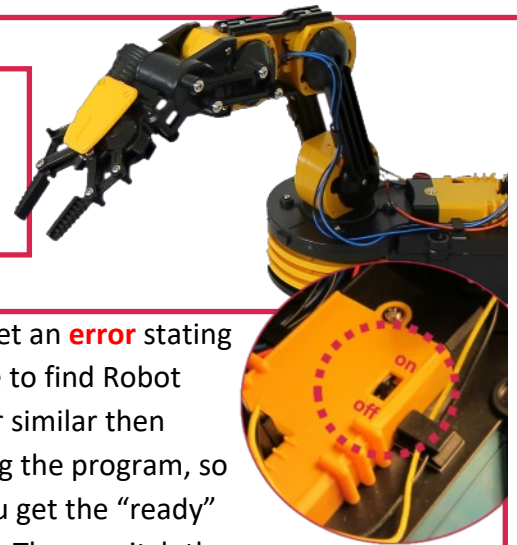


Are you scared of Robots? Well I'm sorry to say, you should be! Unless that is, you know how to program them to do your bidding.

Start a new program in the **Editor [F2]** and enter the code below. When you run it you will be able to control two limbs of the robot arm.



```
CLS
FONTSIZE (4)
INK = RED
PRINT "We are the ROBOTS!"
INK = WHITE
PRINTAT (0, 2); "Press keys to control"
INK = YELLOW
PRINTAT (0, 4); "Left or Right for Body"
PRINTAT (0, 5); "Up or Down for Shoulder"
LOOP
IF SCANKEYBOARD (SCANLEFT) THEN
  ARMBODY (1)
ELSE
  IF SCANKEYBOARD (SCANRIGHT) THEN
    ARMBODY (-1)
  ELSE
    ARMBODY (0)
  ENDIF
ENDIF
ENDIF

IF SCANKEYBOARD (SCANUP) THEN
  ARMSHOULDER (1)
ELSE
  IF SCANKEYBOARD (SCANDOWN) THEN
    ARMSHOULDER (-1)
  ELSE
    ARMSHOULDER (0)
  ENDIF
ENDIF
```

If you get an **error** stating “unable to find Robot Arm” or similar then escaping the program, so that you get the “ready” prompt. Then, switch the arm off at the switch on the arm itself. Turn it back on, and run your program again. It may take a few tries. If this does not work, get in touch at contact@fuze.co.uk

Enter and RUN [F3] the program on the left. Once again, this is a lot of code so you need to be careful not to make any mistakes. However, if you do get an error, try your best to correct it.

Our code starts by setting up our font size and colours we want to use to show the instructions to control the robot on screen.

Notice the **PRINTAT** command. This allows us to position our text exactly where we want it by setting the (X , Y) position.

The main **LOOP** introduces a very useful statement: **SCANKEYBOARD**. This is a great command to check for keys being pressed.

SCANLEFT is a system variable that is either 1 (pressed) or 0 (not pressed) depending on if you press the **LEFT** cursor key. If it **IS** pressed, the robot's body will move to the left. If it is not being pressed, but the **RIGHT** cursor is, then we start moving to the right instead. Lastly, if **neither** the **LEFT** or **RIGHT** key is pressed then we send an **off** signal to the robot's body motor.

Then we do the same for **UP** and **DOWN**, but using the shoulder motor instead.

HACKER CHALLENGE:

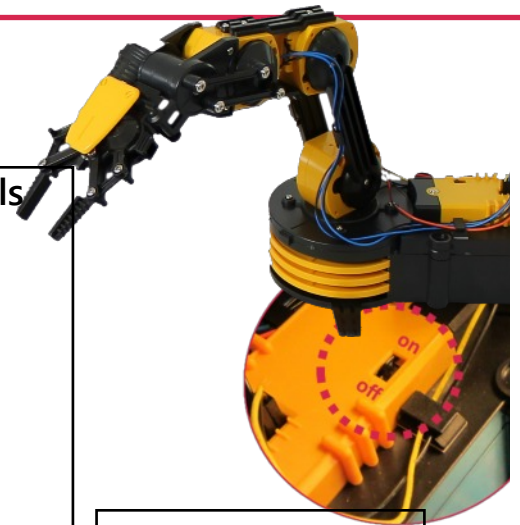
How about adding controls for the rest of the robot's limbs. Notice how each one above has its own section of code beginning with

IF SCANKEYBOARD (SCANkey) THEN ...

The **SCANKEYBOARD** function can be used to detect any key on the keyboard so if you wanted to use **PAGEUP** or **PAGEDOWN** for example you could change **SCANDOWN** to **SCANPAGEUP** or **SCANPAGEDOWN**.

SEE OVERLEAF FOR THE FULL LIST OF SCANKEY CODES

SCANKEYBOARD & Robot reference guide



scanBackspace
 scanTab
 scanClear
 scanReturn
 scanPause
 scanEscape
 scanSpace
 scanExclaim
 scanQuoteDbl
 scanHash
 scanDollar
 scanAmpersand
 scanQuote
 scanLeftParen
 scanRightParen
 scanAsterisk
 scanPlus
 scanComma
 scanMinus
 scanPeriod
 scanSlash
 scan0
 scan1
 scan2
 scan3
 scan4
 scan5
 scan6
 scan7
 scan8
 scan9
 scanColon
 scanSemiColon
 ScanLess
 ScanEquals
 scanGreater
 scanQuestion
 scanAt
 scanLeftBracket
 scanBackSlash
 scanRightBracket
 scanCaret
 ScanUnderscore

scanBackQuote
 scanA
 scanB
 scanC
 scanD
 scanE
 scanF
 scanG
 scanH
 scanI
 scanJ
 scanK
 scanL
 scanM
 scanN
 scanO
 scanP
 scanQ
 scanR
 scanS
 scanT
 scanU
 scanV
 scanW
 scanX
 scanY
 ScanZ
 scanDelete
 scanKP0
 scanKP1
 scanKP2
 scanKP3
 scanKP4
 scanKP5
 scanKP6
 scanKP7
 scanKP8
 scanKP9
 scanKpPeriod
 scanKpDivide
 scanKpMultiply
 scanKpMinus
 scanKpPlus
 ScanKpEnter

ScanKpEquals
 scanUp
 scanDown
 scanRight
 scanLeft
 scanInsert
 scanHome
 scanEnd
 scanPageup
 scanPagedown
 scanF1
 scanF2
 scanF3
 scanF4
 scanF5
 scanF6
 scanF7
 ScanF8
 scanF9
 scanF10
 scanF11
 scanF12
 scanF13
 scanF14
 ScanF15
 scanNumLock
 scanCapsLock
 scanScrolLock

scanRShift
 scanLShift
 scanRCtrl
 scanLCtrl
 scanRAlt
 scanLAlt
 scanRMeta
 scanLMeta
 scanLSuper
 scanRSuper
 scanMode
 scanCompose
 scanHelp
 scanPrint
 scanSysReq
 scanBreak
 scanMenu
 scanPower
 scanEuro

The following Robot Arm commands can be used to control different limbs. Note that a **(1)** sets the limb moving in one direction whereas a **(2)** will set it moving in the opposite one. A **(0)** is used to stop it moving

ARMBODY (x) x can be 1, -1 or 0
ARMSHOULDER (x) x can be 1, -1 or 0
ARMELBOW (x) x can be 1, -1 or 0
ARMWRIST (x) x can be 1, -1 or 0
ARMGRIPPER (x) x can be 1, -1 or 0
ARMLIGHT(x) x can be 1 or 0