PICAXE-08M2 Commands Summary & Help Document

1. Experimenter board (AXE092) symbol definitions

Symbol definitions are very useful for writing better readable programs and are used in the examples here. Symbol definitions do not consume any extra memory, so use them often (including comments/remarks using the ' character)

```
SYMBOL RedLED = C.0                 ' This is the output pin to the red LED
SYMBOL YellowLED = C.1
SYMBOL GreenLED = C.2
SYMBOL SwitchInput = pinC.3
SYMBOL LightSensorInput = C.4
SYMBOL Loudspeaker = C.2
```

2. Variables & Registers

For calculations and storage of small values, 8-bit registers can be used. Larger values can be managed in 16-bit registers (which are combinations of two 8-bit registers*): See LET command (opposite side) for the use of registers in calculations*

- 8-bit registers (valid values: 0 … 255): **b0**, **b1**, **b2**, … , **b27**
- 16-bit registers (valid values: 0 … 65535): **w0**[b1,b0], **w1**[b3,b2], …**w13**[b26,b27]

A register can be given a meaningful name (which should not contain spaces), e.g.:

```
SYMBOL  counter = b2
```

3. Most important commands

All BASIC commands described here are treated in detail in the BASIC manual. This manual can be found under the "Help" section of the Programming Editor.

a. Output commands

HIGH/LOW/TOGGLE pin (,pin, ..): Put one ore more output pin(s) HIGH or LOW:

```
HIGH   RedLED, GreenLED     ' light up red & green LEDs
TOGGLE YellowLED            ' change yellow LED state
```

PLAY tune, option: Play one of the four tunes at the tune pin:

```
 SYMBOL Rudolph_Rednose = 3
PLAY Rudolph_Rednose, 1
```

SOUND channel_nr, (note, duration, note duration, ..) Play sequence of tones with specified duration (in units of 10 millisecs):

```
SOUND  Loudspeaker, (65, 200, 98, 200)   ' Play two notes (2 secs. each)
```

b. Input commands

For testing the level (HIGH or LOW, resp. 1 or 0) of any input pin, see IF THEN ELSE command in the "Program control" section (d) below.

READADC channel_nr, variable: Read the voltage level on an input pin and translate into number between 0 (0V) and max. 255 (equal to battery voltage)

```
SYMBOL LightIntensity = b0
READADC  LightSensorInput, LightIntensity
```

c. Timing related commands

PAUSE millisec: Wait specified number of milliseconds (0 .. 65535)

```
PAUSE  1000      ' wait 1 second = 1000 ms.
```

d. Program control

In order to change the sequence in which a series of commands are executed, a number of control commands are available. These can either make a selection (IF) or repetition of a code part (DO .. LOOP and FOR .. NEXT control commands):

IF condition THEN .. ELSE .. ENDIF: Make a choice based on a condition:

```
IF SwitchInput  = 1 THEN
        HIGH GreenLED
ELSE
        LOW GreenLED
ENDIF
```

DO .. LOOP (WHILE / UNTIL condition): Repeat code while/until conditions holds:

```
DO WHILE SwitchInput = 1
        HIGH YellowLED          ' If switch pressed then light LED
LOOP
LOW YellowLED
```

FOR .. TO .. NEXT: Special compact form of DO .. LOOP when number of repetitions is known

```
FOR  counter = 1 TO 3
        PLAY  Rudolph_Rednose, 1        ' (play song 3 times)
NEXT counter
```

e. Variables etc.

LET variable = expression: Give a variable a value equal to the value of the expression at the right side:

```
SYMBOL counter = b0
SYMBOL square = b1
LET counter = 2
LET counter = counter + 1           ' (counter becomes 3)
LET square = counter * counter      ' (square gets value 3x3=9)
```

f . Other

GOSUB name .. RETURN: Go to a subroutine with the defined name (and with RETURN at its end) and then return to the next command after the calling GOSUB command:

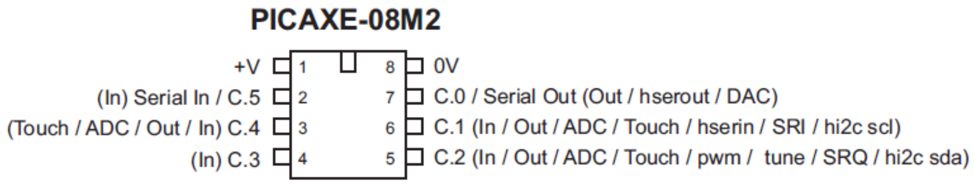
```
GOSUB Blinkleds
(next command) …
…

Blinkleds:
FOR counter = 1 TO 5
        TOGGLE GreenLED
        PAUSE 200
NEXT counter
RETURN
```

RANDOM variable: Generate a random number in the variable

```
SYMBOL GuessWhat = b0
RANDOM GuessWhat
```

4. PICAXE-08M2 pin / channel number definitions


PICAXE-08M2
+V
(In) Serial In / C.5
(Touch / ADC / Out / In) C.4
(In) C.3
1
2
3
4
8
7
6
5
0V
C.0 / Serial Out (Out / hserout / DAC)
C.1 (In / Out / ADC / Touch / hserin / SRI / hi2c scl)
C.2 (In / Out / ADC / Touch / pwm / tune / SRQ / hi2c sda)

Note: **Ignore the pin numbers inside the box**, these have no meaning for programming! Only the channel numbers C.1 – C.5 **outside** are used in programs.

5. HELP, programming the circuit (download) doesn't work !!!

In case of download problems, check using the following steps in sequence:

1. Are the batteries correctly inserted (otherwise: switch off !! and reverse)?
2. Is power switched on, is the programming cable connected?
3. Switch off power to your electronic project circuit, press the Program/Download button in the Programming Editor, and then directly switch on the power again, now a download should start directly.
4. Is the correct serial port selected (always to be checked when the Programming Editor is started, always connect the cable before starting the editor)!
    a. In the Workspace Explorer window (Settings tab) go to the "COM Port" section, select the AXE027 PICAXE USB port from the list.
    b. If it was not on the list, click on "Refresh COM ports", then it should be in the list again. If not, the required USB-driver was not installed.
    c. Install the USB-drivers for the AXE027 download cable (see picaxe website for instructions) and start at step 1 again.
5. Do a "Check PICAXE type connected" operation in the Workspace Explorer/ PICAXE Type section, check also that PICAXE-08M2 (or the device you use) is selected, this should return a firmware revision number.
6. Check battery voltage, still reasonably charged (> 2.4 V)?
7. Call The Local Expert or – if not in a hurry - ask on the Picaxe forum