

Pin Diagram of the BASIC Stamp

Pin Designations:

Ground is also called Vss.

Vcc is +5V, even though we use a 9v battery. **Vcc** is also called Vdd.

Reset is active low, and is the same as pushing the reset button on the BOE.

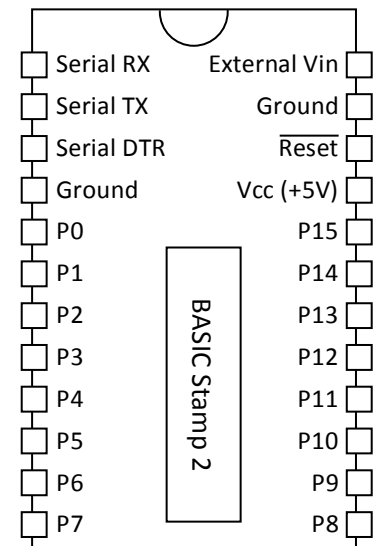
External Vin allows the chip to be powered from a separate source than the board of education. The Stamp will convert this (6v to 12v) power supply into a +5v Vcc. It CANNOT be used when a battery is connected. Don't touch it.

Sout: When the Stamp is connected to a PC, this pin talks to the RX pin of the serial port running the STAMP.

Sin: When the Stamp is connected to a PC, this pin talks to the TX pin of the serial port running the STAMP.

Atn: When the Stamp is connected to a PC, this pin talks to the DTR pin of the serial port running the STAMP.

P0 through P15: Programmable I/O pins.



Board of Education

Allows communication with PC through serial cable or USB cable.

Requires a power supply (e.g., a 9V battery).

The terminal strips allow easy wiring access to the sixteen pins of the stamp.

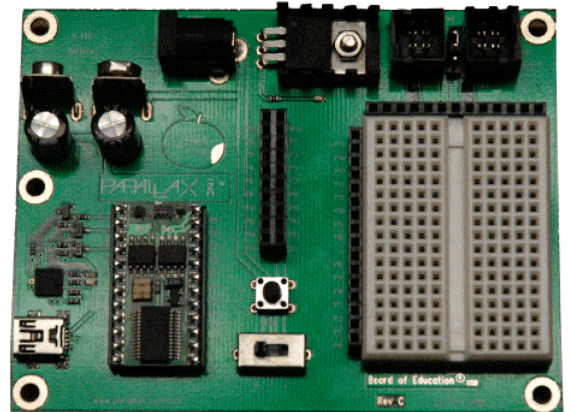
The Stamp can be used with the BOE after it has been programmed.

Has a physical reset button to restart the stamp's current program.

The power switch should be off (left) when not in use, otherwise, the LED will drain the battery. The on position is in the middle. The right most position allows use of other features we will not use.

Getting Started:

- Attach the battery
- Put the switch in the middle (1) position
- Run the BASIC Stamp Editor
- Connect ground (Vss) of the BOE to ground for your PAD-234.



Common PBASIC Commands

NOTE: PBASIC is NOT case sensitive! Q is the same as q.

{STAMP BS2} Specifies that we are using the version 2 Basic Stamp. REQUIRED for all programs!

{PBASIC 2.5} Specifies that we are using PBASIC version 2.5. REQUIRED for all programs!

Allows the user to write programming comments. Also used for global directives, as above.

END: Put the Stamp into low-power mode. Use at the end of all programs!

PAUSE 200: Program will wait for specified time in milliseconds.

Variables

Q VAR Byte: Create an integer variable name (in this case, Q)

Allowed types are Bit (0 to 1), Nib (4 bits, 0 to 15), byte (8 bits, 0 to 255), and Word (16 bits, 0 to 65545)

A VAR Byte(20): create a twenty element array called A.

Q = (value or expression) : Assign a value to a variable. Value can be (decimal), %(binary), \$(hexadecimal), or "(ASCII character)" and is truncated to match the size of Q.

M CON 42 : Define a constant (in this case, M) that makes your program easier to read, but isn't a variable.

Bob PIN 9: Assign the alias "Bob" to Pin 9. Bob can be understood as the number 9 (OUTPUT Bob), an alias to OUT9 (Bob=1), or an alias to IN9 (Q=Bob+3), as determined by context.

Common PBASIC Commands Continued

External Interaction

DEBUG: Instructs the stamp to send a message to the PC.

- DEBUG 65 → send ASCII character 65 (“A”) to the display
- DEBUG DEC 65 → send the integer 65 to the display
- DEBUG “hi there” → send the message “hi there” to the display.
- DEBUG CR → send a carriage return to the display. Same as ASCII 13.
- DEBUG 65, 66, 67 → send multiple characters to the display (in this case, “ABC”).
- DEBUG ?Q → display the value of the declared variable “Q”.
- DEBUG HOME → moves the display cursor to the upper left corner.
- DEBUG CLS → clear the display screen
- DEBUGIN DEC q → waits for user to type in a value for the variable q.

INX : Variable containing the value on pin(s) *X* (at the time it is referenced). Read only!

OUTX : Variable containing the value in output latch *X*.

DIRX : Variable controlling whether the output latches are connected to the pins: 1 for yes (pin is an output) or 0 for no (pin is an input). *X* can be **S** (the whole word), **H**, **L** (high and low bytes), **A**, **B**, **C**, **D** (nibbles low to high), or **0–15** (for individual pins)

INPUT 4 : Specify that pin 4 is an input. Same as DIR4 = 0, except the pin number can be an expression.

OUTPUT 12 : Specify that pin 12 is an output. Same as DIR12 = 1.

HIGH 5 ; **LOW 5** : Asserts a pin (in this case, 5) as high or low, and then sets it as an output (if not already).

PULSEOUT 3, 150 : Assert a pulse of 150*2 microseconds to pin 3, with polarity opposite its current value

FREQOUT 3, 500, 2240 : On pin 3, output a sine wave of frequency 2240 Hz for a duration 500ms.

Calculation

+ – * / : 16 bit integer math, executed left to right (no precedence rules, but parentheses allowed)

A*/B : Compute A*B/256. Allows you to multiply by a non-integer F by setting B=F*256.

AB** : Compute A*B/65536. Allows you to multiply by a fraction F (less than one) by setting B=F*65536.

A//B : Compute the remainder of the division A / B. AKA, the modulus operator.

~ & | ^ << >> : Bit level operators, bitwise Boolean and shift (fill with 0)

=, <>, >, <, <=, >= : Relational operators, only use in conditional statement

NOT, AND, OR, XOR : Logical operators (i.e., not bitwise), only use in conditional statement

RANDOM q: Assign a random value to q, using the range allowed for that type.

Program control

DO : Establish the starting point for a repeating loop. Use for all designs used without the BOE.

LOOP : Return to DO.

LOOP UNTIL → specify a condition for loop termination.

FOR q = 4 TO 13 STEP 3: Begin a loop, assigning 4 to q initially, and incrementing q by 3 each cycle.

NEXT : Begin a new cycle of the associated FOR loop, or exit the loop if the end is reached.

EXIT : Force an exit from the innermost loop (either a DO or a FOR).

IF (Q = 1) THEN ... ELSIF () THEN ... ELSE ... ENDIF : Conditional branch.

Note that “=” is used for both assignment and comparison! ELSIF and ENDIF only for multi-line.

ComeHere : Defines a label for a point in the program. Note that the colon is part of the code

GOTO ComeHere : Move execution to a label. The colon is not part of this command