

\ Dictionary	C!	CSET	CCLR	@	LSHIFT	DUP	DROP	MS	
\	AND	OR	XOR	INVERT	RSHIFT	BEGIN	UNTIL	IF	
\	:	;	.S	.	U.	KEY?	?	HEX	DECIMAL

HEX \ change communication to hexadecimal

11 22 33 \ put some data on to the stack
 .S \ and check it is there

00 2E c! \ set port 2 to just be GPIO, no internal specials used

80 2A cset \ set bit 7 DDR to output. Replace 80 by 40/20/10 on output bits

80 29 cset \ set bit 7 HIGH

80 29 cclr \ set bit 7 LOW

variable OUTPUT \ define a variable OUTPUT

77 OUTPUT ! \ load the variable with 77

Output @ . \ display the data in OUTPUT

: OUT OUTPUT @ 4 lshift 29 c! ; \ Output variable to LED

0F 2A cclr \

0F 2F cset \

0F 29 cset \

: INIT 00 2e c! F0 2a C! 0F 2F cset 0F 29 cset ;

: 4SHIFTL 4 LSHIFT 0F OR ;

: INTOOUT INIT BEGIN 28 C@ 4SHIFTL 29 C! KEY? UNTIL ;

1000 ms 80 29 cset \.....

500 ms 80 29 cclr \.....

: delay begin dup ms 80 29 cset dup ms 80 29 cclr key? until ; \.....

DECIMAL \ set to decimal for easier calculations \.....

: DIT 80 29 cset 300 ms 80 29 cclr 100 ms ; \ LED on 300ms, 100 off .

: DAA 80 29 cset 900 ms 80 29 cclr 100 ms ; \ LED on 900ms, 100 off

: SP 300 ms ; \ short pause between letters

: LP 999 ms ; \ long pause between words

: MorseSOS begin dit dit dit sp daa daa daa sp dit dit dit sp key? until ; .

: AND01 BEGIN 28 c@ DUP 1 rshift AND 01 AND 4shiftrl 29 c! key? until ; ..

: OR01 BEGIN 28 C@ DUP 1 RSHIFT OR 01 AND 4SHIFTL 29 C! KEY? UNTIL ; \...

: XOR01 BEGIN 28 C@ DUP 1 RSHIFT XOR 01 AND 4SHIFTL 29 C! KEY? UNTIL ; \..

: INVERT0 BEGIN 28 C@ INVERT 01 AND 4SHIFTL 29 C! KEY? UNTIL ; \...

: ToKB Decimal begin 1 . 1000 ms key? until ; \.send 1 to show each 1 sec

: INITADC 00 2E c! F0 2A c! 0F 2F cset 0F 29 cset ; \.....

variable ADCV

: ADC initadc 0 ADCV ! BEGIN 1 ADCV +! 1 28 cget UNTIL ADCV @ . ; \.

: ??? 2E @ . 2A @ . 28 @ . OUTPUT @ . 29 @ . base @ >r hex .S r> base ! ;

\ ??? 0 F0 FF 77 FF <1> -3E22 ok - GPIO DDR INPUTS vOUTPUT LEDs STACK