1802 Instruction Set

I (1st byte) / **N** (Second Instruction Byte)

I \ N	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	IDL	LDN - Load D via Register N														
1	INC N - Increment Register N															
2	DEC N - Decrement Register N															
3	BR	BQ	BZ	BDF	B1	B2	B3	B4	SKP	BNQ	BNZ	BNF	BN1	BN2	BN3	BN4
4	LDA N - Load D via Register N and Advance															
5	STR N - Store D via Register N															
6	IRX	OUT N (Output 1-7)							DBG	INP N (Input 1-7)						
7	RET	DIS	LDXA	STXD	ADC	SDB	SHRC	SMB	SAV	MARK	REQ	SEQ	ADCI	SDBI	SHLC	SMBI
8	GLO N - Get Low Byte from Register N, Put in D															
9	GHI N - Get High Byte from Register N, Put in D															
A	PLO N - Put D into Low Byte of Register N															
B	PHI N - Put D into High Byte of Register N															
C	LBR	LBQ	LBZ	LBDF	NOP	LSNQ	LSNZ	LSNF	LSKP	LBNQ	LBNZ	LBNF	LSIE	LSQ	LSZ	LSDF
D	SEP N - Set P to N															
E	SEX N - Set X to N															
F	LDX	OR	AND	XOR	ADD	SD	SHR	SM	LDI	ORI	ANI	XRI	ADI	SDI	SHL	SMI

Instruction Type: Branches & Skips | I/O | Register Operations | Arithmetic & Logic | Control

Sparkfun Qwiic 3x4 Keypad

Key Values:

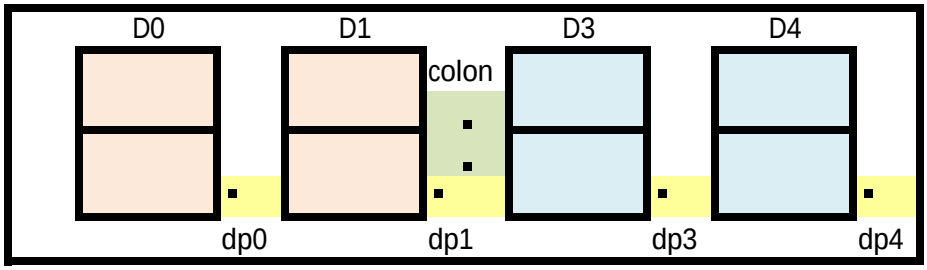
1	2	3
4	5	6
7	8	9
* Shift	0	# Input

Shifted Keys:

A	B	C
D	E	F
Load	Run Wait	Memory Protect
Debug	Clear	Hold Input

Keystrokes	Result
0-9	Enter hex digit 0-9
#	Simulate Input key press (EF4=1)
*, 1-6	Enter hex digit A-F
*, 7	Load mode
*, 8	Toggle Run / Wait mode
*, 9	Memory Protect on/off
*, 0	Clear mode
*, #	Simulate Hold / Release Input key (Toggle EF4 = 1/0)
,	Debug: toggle single-step in wait/run mode. Back up one byte, in load mode.

LED Backpack



D0, D1 - Address Byte in Hex (off in run mode)
D3, D4 - Data Byte / Opcode in Hex
dp0 - Memory Protect (on) / Write Enable (off)
dp1 - Single Step (on)
dp3 - Load (on)
dp4 - Run (on) / Wait (off)
colon - In run mode: CPU Idle (on) / CPU Run (off)

Mnemonic	Name	Opcode	Cycles
ADC	Add with Carry	74	2
ADCI bb	Add with Carry Immediate	7C bb	2
ADD	Add	F4	2
ADI bb	Add Immediate	FC bb	2
AND	Logical AND	F2	2
ANI bb	AND Immediate	FA bb	2
B1 aa	Branch on External Flag 1	34 aa	2
B2 aa	Branch on External Flag 2	35 aa	2
B3 aa	Branch on External Flag 3	36 aa	2
B4 aa	Branch on External Flag 4	37 aa	2
BDF aa	Branch if DF is 1	33 aa	2
BN1 aa	Branch on Not External Flag 1	3C aa	2
BN2 aa	Branch on Not External Flag 2	3D aa	2
BN3 aa	Branch on Not External Flag 3	3E aa	2
BN4 aa	Branch on Not External Flag 4	3F aa	2
BNF aa	Branch if DF is 0	3B aa	2
BNQ aa	Branch if Q is off	39 aa	2
BNZ aa	Branch on Not Zero	3A aa	2
BQ aa	Branch if Q is on	31 aa	2
BR aa	Branch unconditionally	30 aa	2
BZ aa	Branch on Zero	32 aa	2
DBG	Debug breakpoint - Halt enter Wait mode	68	2
DEC r	Decrement Register	2r	2
DIS	Return and Disable Interrupts	71	2
GHI r	Get High byte of Register	9r	2
GLO r	Get Low byte of Register	8r	2
IDL	Idle	00	2
INC r	Increment Register	1r	2
INP p	Input to memory and D (for p = 9 to F)	6p	2
IRX	Increment R(X)	60	2
LBDF aaaa	Long Branch if DF is 1	C3 aaaa	3
LBNF aaaa	Long Branch if DF is 0	CB aaaa	3
LBNQ aaaa	Long Branch if Q is off	C9 aaaa	3
LBNZ aaaa	Long Branch if Not Zero	CA aaaa	3
LBQ aaaa	Long Branch if Q is on	C1 aaaa	3
LBR aaaa	Long Branch unconditionally	C0 aaaa	3
LBZ aaaa	Long Branch if Zero	C2 aaaa	3
LDA r	Load D and Advance	4r	2
LDI bb	Load D Immediate	F8 bb	2
LDN r	Load D via N (for r = 1 to F)	0r	2
LDX	Load D via R(X)	F0	2
LDXA	Load D via R(X) and Advance	72	2
LSDF	Long Skip if DF is 1	CF	3
LSIE	Long Skip if Interrupts Enabled	CC	3
LSKP	Long Skip	C8	3
LSNF	Long Skip if DF is 0	C7	3
LSNQ	Long Skip if Q is off	C5	3
LSNZ	Long Skip if Not Zero	C6	3

Mnemonic	Name	Opcode	Cycles
LSQ	Long Skip if Q is on	CD	3
LSZ	Long Skip if Zero	CE	3
MARK	Save X and P in T	79	2
NOP	No Operation	C4	3
OR	Logical OR	F1	2
ORI bb	OR Immediate	F9 bb	2
OUT p	Output from memory (for p = 1 to 7)	6p	2
PHI r	Put D in High byte of register	Br	2
PLO r	Put D in Low byte of register	Ar	2
REQ	Reset Q	7A	2
RET	Return	70	2
SAV	Save T	78	2
SD	Subtract D from memory	F5	2
SDB	Subtract D from memory with Borrow	75	2
SDBI bb	Subtract D with Borrow, Immediate	7D bb	2
SDI bb	Subtract D from memory Immediate byte	FD bb	2
SEP r	Set P	Dr	2
SEQ	Set Q	7B	2
SEX r	Set X	Er	2
SHL	Shift D Left	FE	2
SHLC	Shift D Left with Carry	7E	2
SHR	Shift D Right	F6	2
SHRC	Shift D Right with Carry	76	2
SKP	Skip one byte	38	2
SM	Subtract Memory from D	F7	2
SMB	Subtract Memory from D with Borrow	77	2
SMBI bb	Subtract Memory with Borrow, Immediate	7F bb	2
SMI bb	Subtract Memory from D, Immediate	FF bb	2
STR r	Store D into memory	5r	2
STXD	Store D via R(X) and Decrement	73	2
XOR	Exclusive OR	F3	2
XRI bb	Exclusive OR, Immediate	FB bb	2

Legend:

aa	8-bit Address
aaaa	16-bit Address
bb	8-bit Data Byte
p	Port 1-7 Output; 9-F Input
r	Register (0 - F)

Instruction Type:

Branches & Skips
I/O
Register Operations
Arithmetic & Logic
Control

Opcode	Mnemonic	Name	Cycles
00	IDL	Idle	2
0r	LDN r	Load D via N (for r = 1 to F)	2
1r	INC r	Increment Register	2
2r	DEC r	Decrement Register	2
30 aa	BR aa	Branch unconditionally	2
31 aa	BQ aa	Branch if Q is on	2
32 aa	BZ aa	Branch on Zero	2
33 aa	BDF aa	Branch if DF is 1	2
34 aa	B1 aa	Branch on External Flag 1	2
35 aa	B2 aa	Branch on External Flag 2	2
36 aa	B3 aa	Branch on External Flag 3	2
37 aa	B4 aa	Branch on External Flag 4	2
38	SKP	Skip one byte	2
39 aa	BNQ aa	Branch if Q is off	2
3A aa	BNZ aa	Branch on Not Zero	2
3B aa	BNF aa	Branch if DF is 0	2
3C aa	BN1 aa	Branch on Not External Flag 1	2
3D aa	BN2 aa	Branch on Not External Flag 2	2
3E aa	BN3 aa	Branch on Not External Flag 3	2
3F aa	BN4 aa	Branch on Not External Flag 4	2
4r	LDA r	Load D and Advance	2
5r	STR r	Store D into memory	2
60	IRX	Increment R(X)	2
6p	OUT p	Output from memory (for p = 1 to 7)	2
68	DBG	Debug breakpoint - Halt enter Wait mode	2
6p	INP p	Input to memory and D (for p = 9 to F)	2
70	RET	Return	2
71	DIS	Return and Disable Interrupts	2
72	LDXA	Load D via R(X) and Advance	2
73	STXD	Store D via R(X) and Decrement	2
74	ADC	Add with Carry	2
75	SDB	Subtract D from memory with Borrow	2
76	SHRC	Shift D Right with Carry	2
77	SMB	Subtract Memory from D with Borrow	2
78	SAV	Save T	2
79	MARK	Save X and P in T	2
7A	REQ	Reset Q	2
7B	SEQ	Set Q	2
7C bb	ADCI bb	Add with Carry Immediate	2
7D bb	SDBI bb	Subtract D with Borrow, Immediate	2
7E	SHLC	Shift D Left with Carry	2
7F bb	SMBI bb	Subtract Memory with Borrow, Immediate	2
8r	GLO r	Get Low byte of Register	2
9r	GHI r	Get High byte of Register	2
Ar	PLO r	Put D in Low byte of register	2
Br	PHI r	Put D in High byte of register	2

Opcode	Mnemonic	Name	Cycles
C0 aaaa	LBR aaaa	Long Branch unconditionally	3
C1 aaaa	LBQ aaaa	Long Branch if Q is on	3
C2 aaaa	LBZ aaaa	Long Branch if Zero	3
C3 aaaa	LBDF aaaa	Long Branch if DF is 1	3
C4	NOP	No Operation	3
C5	LSNQ	Long Skip if Q is off	3
C6	LSNZ	Long Skip if Not Zero	3
C7	LSNF	Long Skip if DF is 0	3
C8	LSKP	Long Skip	3
C9 aaaa	LBNQ aaaa	Long Branch if Q is off	3
CA aaaa	LBNZ aaaa	Long Branch if Not Zero	3
CB aaaa	LBNF aaaa	Long Branch if DF is 0	3
CC	LSIE	Long Skip if Interrupts Enabled	3
CD	LSQ	Long Skip if Q is on	3
CE	LSZ	Long Skip if Zero	3
CF	LSDF	Long Skip if DF is 1	3
Dr	SEP r	Set P	2
Er	SEX r	Set X	2
F0	LDX	Load D via R(X)	2
F1	OR	Logical OR	2
F2	AND	Logical AND	2
F3	XOR	Exclusive OR	2
F4	ADD	Add	2
F5	SD	Subtract D from memory	2
F6	SHR	Shift D Right	2
F7	SM	Subtract Memory from D	2
F8 bb	LDI bb	Load D Immediate	2
F9 bb	ORI bb	OR Immediate	2
FA bb	ANI bb	AND Immediate	2
FB bb	XRI bb	Exclusive OR, Immediate	2
FC bb	ADI bb	Add Immediate	2
FD bb	SDI bb	Subtract D from memory Immediate byte	2
FE	SHL	Shift D Left	2
FF bb	SMI bb	Subtract Memory from D, Immediate	2

Legend:

aa	8-bit Address
aaaa	16-bit Address
bb	8-bit Data Byte
p	Port 1-7 Output; 9-F Input
r	Register (0 - F)

Instruction Type:

Branches & Skips
I/O
Register Operations
Arithmetic & Logic
Control