

Instruction set

	code	operand	notes
basic instruction	FW	1 .. 9, 0	unit time action, 0: random
	BK	1 .. 9, 0	unit time action, 0: random
	LT (Left Turn)	1 .. 9, 0	unit time action, 0: random
	RT (Right Turn)	1 .. 9, 0	unit time action, 0: random
	BEEP, LED	1 .. 9, 0	scale, 0: LED mode
flow instruction	FOR	1 .. 9, 0	repeat time, 0: ever
	NEXT	-	terminal of FOR and WHILE
	IF	SensorID+ON/OFF	Sensor condition
	ELSE	-	ELSE
	ENDIF	-	terminal of IF statement
	RUN/STOP		program auto save
	Reset		memory clear

State transition

FOR	↑	NEXT	P0
↺	BEEP LED	↻	P1
IF	↓	ENDIF	P2
RUN/ STOP	ELSE	Reset	

state 0

Sonar 10cm	Compass	Sonar 30cm	P0 (digital)
A button	Brightness	B button	P1 (digital)
Sonar 170cm	Accel	Sonar 90cm	P2 (analog)
	Temperature		

state 2

micro:bit FIT firmware ver. 0.A

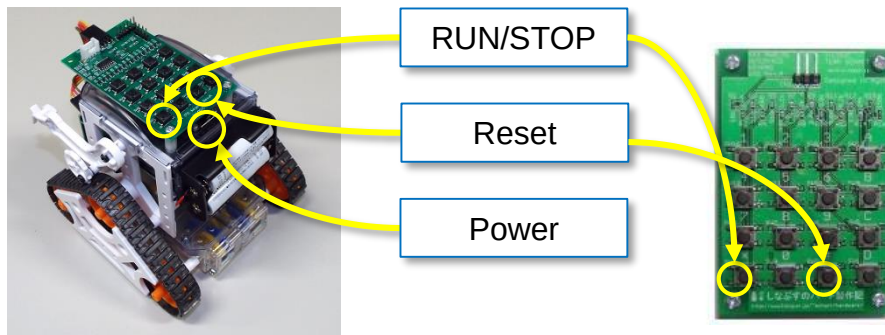
1	2	3
4	5	6
7	8	9
*	0	#

state 1

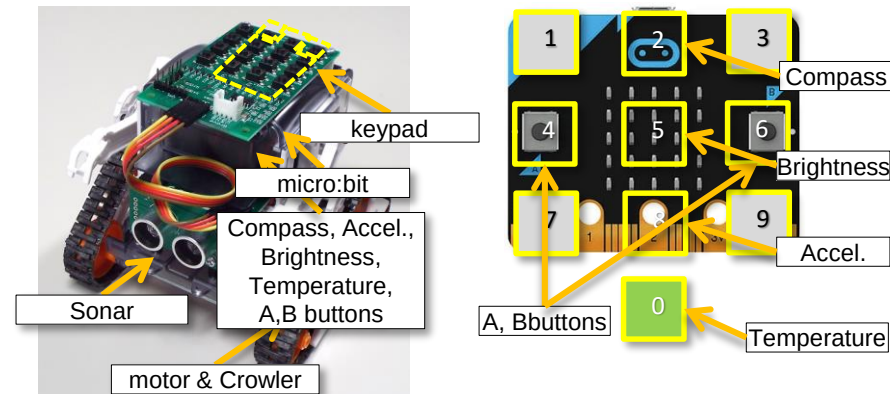
	ON	
	OFF	

state 3

Power, Reset, RUN/STOP



Layout of sensors



Sample codes

```
// Basic
FW [2] 5
BK [8] 5
LT [4] 4
RT [6] 4
BP [5] 5

// LED
LED [5] 0
1~9
0
```

```
// U turn
FW [2] 9
LT [4] 8
FW [2] 9
```

```
// DO-RE-MI
BP [5] 1
BP [5] 2
BP [5] 3
```

```
// Roulette
RT [6] 0
RT [6] 0
RT [6] 0
RT [6] 0
```

```
// "random"
// cannot be
// used in BEEP
```

```
// Draw square
FW 5
RT 4
FW 5
RT 4
FW 5
RT 4
FW 5
RT 4
```

```
// Repeat
FOR 4 1 4
FW 5 2 5
RT 4 6 4
NEXT 3

// "FOR 0" means
// inf. loop
```

```
// Show version
[RESET]+[0]+[0]
```

```
// Show coding times
[RESET]+[0]+[1]
```

```
// Show total code number
[RESET]+[0]+[4]
```

```
// Show code
[RESET]+[0]+[2]
```

```
// data logging
[RESET]+[0]+[5]
```

```
// Output log data
[RESET]+[0]+[2]
```

```
// Radio comm.
[RESET]+[0]+[3]
```

```
// Radio ( sender)
[RESET]+[0]+[2]
```

```
// Obstacle avoid with sonar
FOR ever 1 0
IF 10cmSonar ON 7 1 2
RT 2 6 2
ENDIF 9
NEXT 3
```

The ranges detected by sonar are 10cm, 30cm, 90cm and 170cm..

```
// Beeping if bright
FOR ever 1 0
IF Bright ON 7 5 2
BP So 5 5
ENDIF 9
NEXT 3
```

If you cover the robot or micro:bit with clothes, micro:bit beeps.
The threshold of brightness is automatically set.

```
// Robot goes to the South
FOR 0 1 0
IF Compass ON 7 2 2
FW 1 2 1
ELSE 0
LT 1 4 1
ENDIF 9
NEXT 3
```

Since the micro:bit is set upside down in the robot, this code finds the South.

```
// Tracking by use of sonar
FOR 0 1 0
IF 30cmSonar OFF 7 3 8
LeftTurn 1 4 1
ELSE 0
IF 10cmSonar OFF 7 1 8
FW 1 2 1
ENDIF 9
ENDIF 9
NEXT 3
```

It searches for something within 30 cm in front of it, and when it finds it, it approaches it.

```
// Beeping if the South
FOR 0 1 0
IF Compass ON 7 2 2
BP So 5 5
ENDIF 9
NEXT 3
```

Compass is calibrated automatically while you use the micro:bit.

```
// Beeping if tilted 45 deg over
FOR 0 1 0
IF Accel ON 7 8 2
BP So 5 5
ENDIF 9
NEXT 3
```

Measures gravity and acceleration; turns ON when tilted more than 45 degrees and when acceleration changes more than 0.2G.

```
// Temperature sensor
IF Temp. sensor ON 7 0 2
ENDIF 9
```

Show temperature by beeping as binarized decimal number

```
// WHILE statement
IF 10cmSonar OFF 7 1 8
FW 1 2 1
NEXT 3
```

The terminal of WHILE statement is NEXT command.