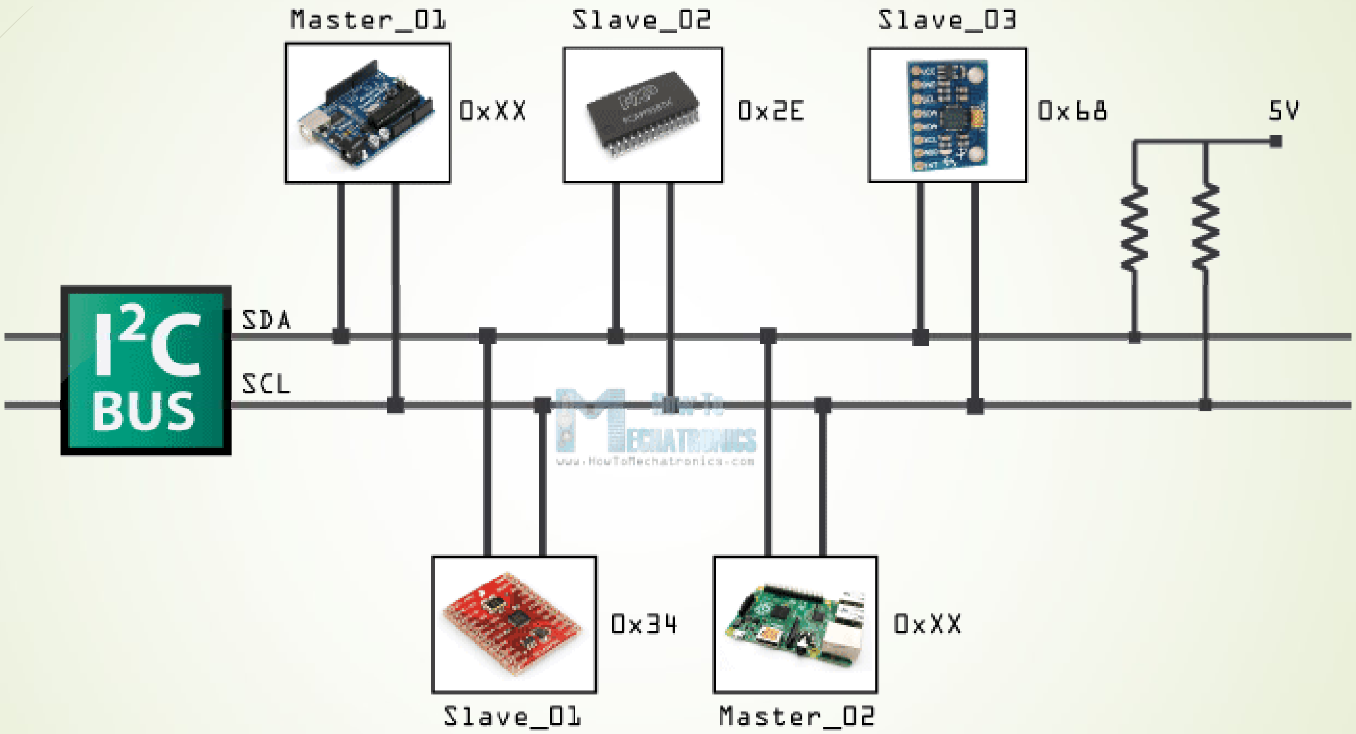
BMS 302 MİKROKONTROLCÜLER

I2C HABERLEŞMESİ

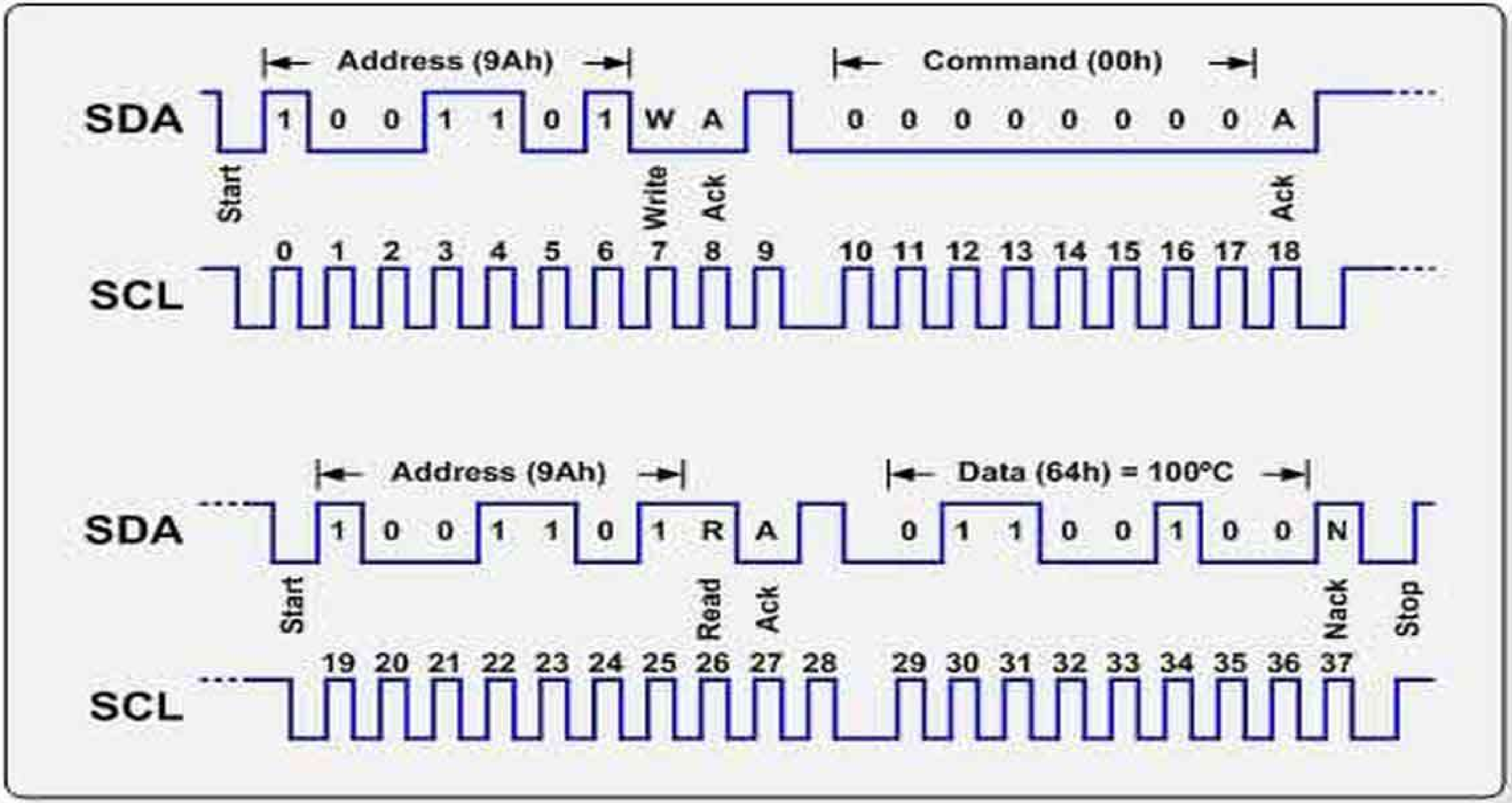
Prof.Dr. Mutlu AVCI

Çukurova Üniversitesi Biyomedikal Mühendisliği Bölümü

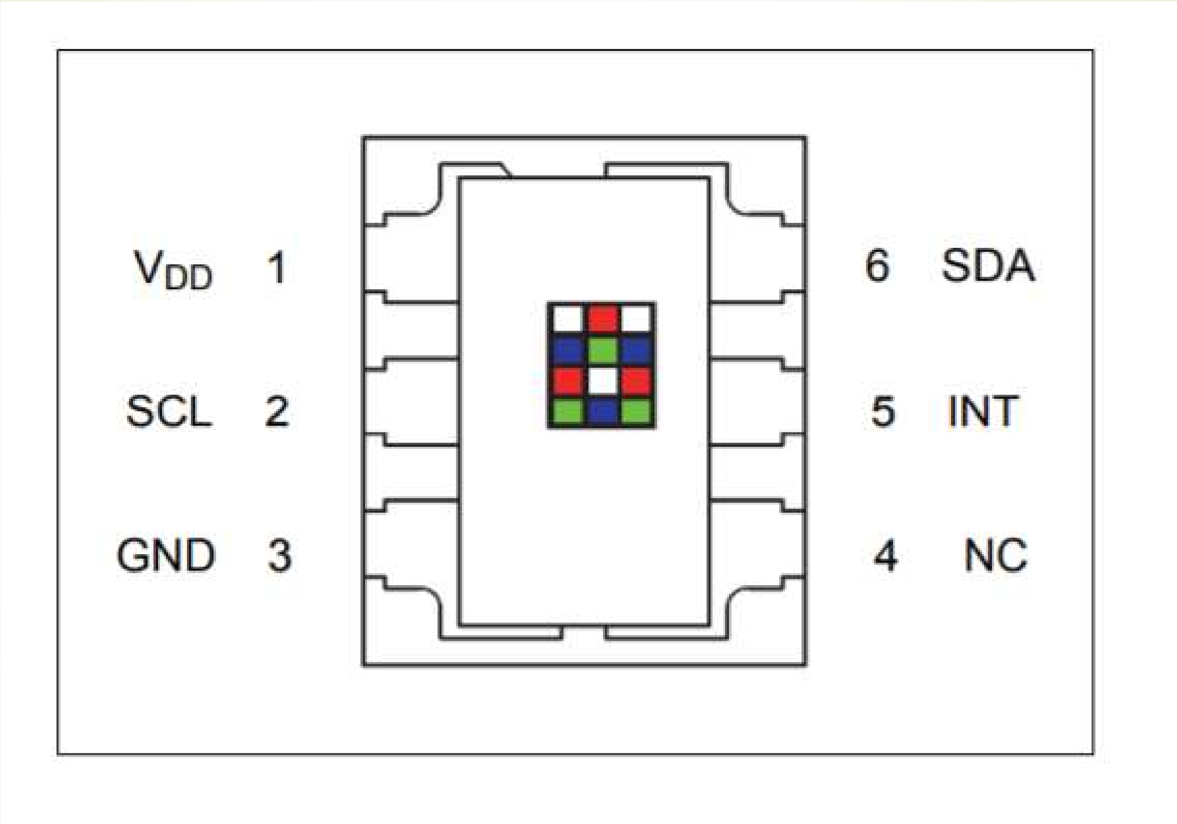
Bahar 2020



Master_01
0xXX
Slave_02
0x2E
Slave_03
0x68
5V
I2C BUS
SDA
SCL
Slave_01
0x34
Master_02
0xXX



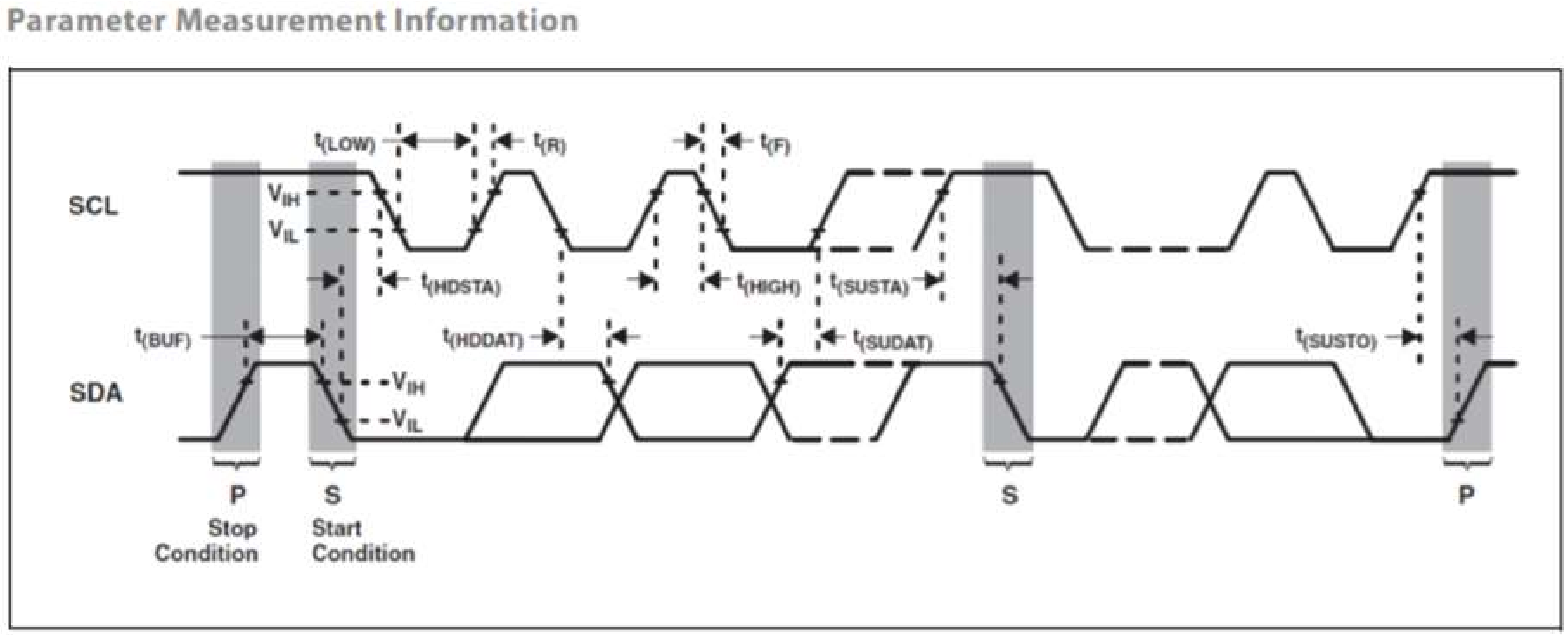
Address (9Ah)
Command (00h)
SDA
1 0 0 1 1 0 1 W A
0 0 0 0 0 0 0 0 A
Start
Write
Ack
Ack
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
SCL
Address (9Ah)
Data (64h) = 100°C
SDA
1 0 0 1 1 0 1 R A
0 1 1 0 0 1 0 0 N
Start
Read
Ack
Nack
Stop
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
SCL

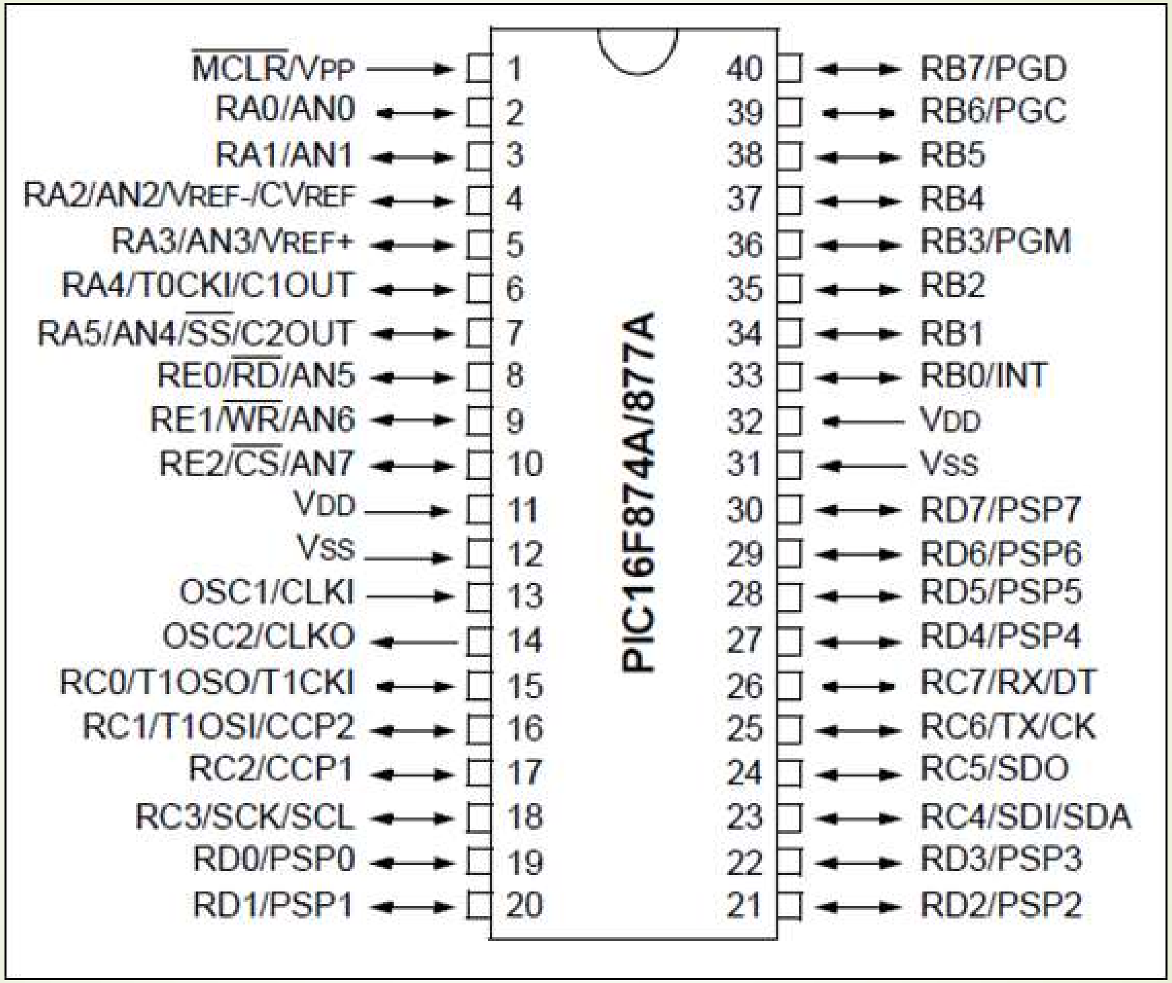

VDD 1
SCL 2
GND 3
6 SDA
5 INT
4 NC

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{DD}	Supply voltage	(I^2C $V_{BUS} = V_{DD}$)	2.7	3	3.6	V
		(I^2C V_{BUS} = 1.8 V)	2.7	3	3.3	
T_A	Operating free - air temperature		-30		70	ºC

Parameter	Description	Min	Typ	Max	Units
$f_{(SCL)}$	Clock frequency (I^2C only)	0		400	kHz
$t_{(BUF)}$	Bus free time between start and stop condition	1.3			µs
$t_{(HDSTA)}$	Hold time after (repeated) start condition. After this period, the first clock is generated.	0.6			µs
$t_{(SUSTA)}$	Repeated start condition setup time	0.6			µs
$t_{(SUSTO)}$	Stop condition setup time	0.6			µs
$t_{(HDDAT)}$	Data hold time	0		0.9	µs
$t_{(SUDAT)}$	Data setup time	100			ns
$t_{(LOW)}$	SCL clock low period	1.3			µs
$t_{(HIGH)}$	SCL clock high period	0.6			µs
t_F	Clock/data fall time			300	ns
t_R	Clock/data rise time			300	ns
C_i	Input pin capacitance			10	pF

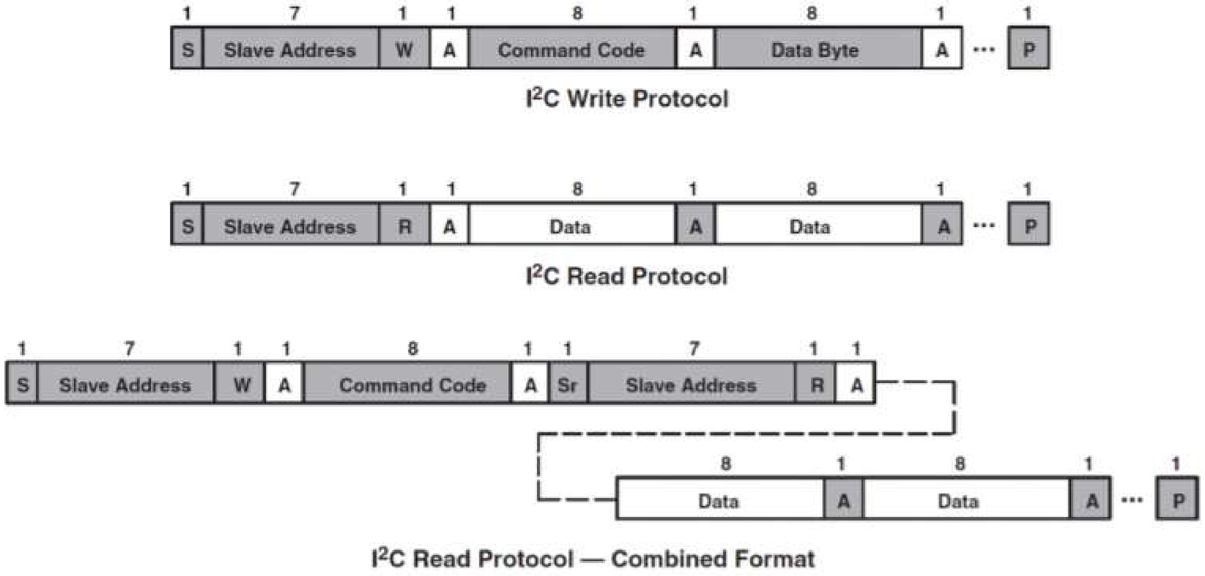
Timing Diagram



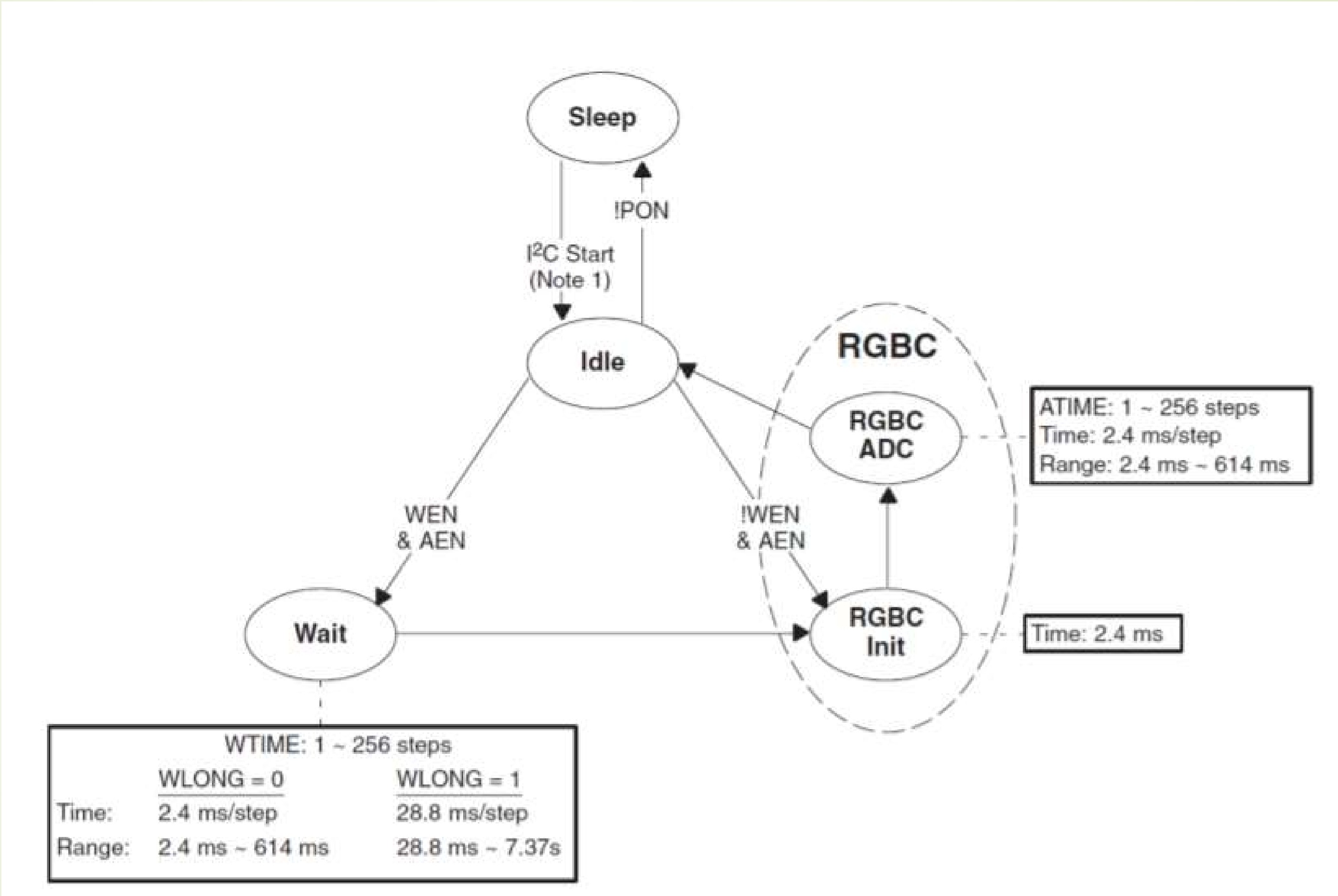


PIC16F874A/877A
MCLR/VPP 1
RA0/AN0 2
RA1/AN1 3
RA2/AN2/VREF-/CVREF 4
RA3/AN3/VREF+ 5
RA4/T0CKI/C1OUT 6
RA5/AN4/SS/C2OUT 7
RE0/RD/AN5 8
RE1/WR/AN6 9
RE2/CS/AN7 10
VDD 11
VSS 12
OSC1/CLKI 13
OSC2/CLKO 14
RC0/T1OSO/T1CKI 15
RC1/T1OSI/CCP2 16
RC2/CCP1 17
RC3/SCK/SCL 18
RD0/PSP0 19
RD1/PSP1 20
40 RB7/PGD
39 RB6/PGC
38 RB5
37 RB4
36 RB3/PGM
35 RB2
34 RB1
33 RB0/INT
32 VDD
31 VSS
30 RD7/PSP7
29 RD6/PSP6
28 RD5/PSP5
27 RD4/PSP4
26 RC7/RX/DT
25 RC6/TX/CK
24 RC5/SDO
23 RC4/SDI/SDA
22 RD3/PSP3
21 RD2/PSP2

I²C Protocols



A	Acknowledge (0)	**Sr**	Repeated Start Condition
N	Not Acknowledged (1)	**W**	Write (0)
P	Stop Condition	**···**	Continuation of Protocol
R	Read (1)	▩	Master - to - Slave
S	Start Condition	☐	Slave - to - Master


Sleep
!PON
I2C Start
(Note 1)
Idle
RGBC
RGBC
ADC
ATIME: 1 ~ 256 steps
Time: 2.4 ms/step
Range: 2.4 ms ~ 614 ms
WEN
& AEN
!WEN
& AEN
Wait
RGBC
Init
Time: 2.4 ms
WTIME: 1 ~ 256 steps
WLONG = 0
WLONG = 1
Time: 2.4 ms/step 28.8 ms/step
Range: 2.4 ms ~ 614 ms 28.8 ms ~ 7.37s

CCS C ile I2C

#use i2c(sabit, sabit, sabit,…)

Sabitler:

MULTI_MASTER	Çoklu Master Modu
MASTER	
SLAVE	
SCL=pin	
SDA=pin	
FORCE_HW	
FORCE_SW	

SLOW	I2C yavaş haberleşecek (SLOW=100000)
FAST	I2C hızlı haberleşecek (FAST=400000)
ADDRESS	Slave adres değeri
RESTART_WDT	
STREAM=isim	
I2C1	
I2C2	

Örnek:

#use i2c(master, sda=pin_B4, scl=pin_B3, fast=400000)

Örnek:

#use i2c(master,sda=pin_C4, scl=pin_C3, fast, FORCE_HW, STREAM=iletişim_2)

I2C_START() veya I2C_START(STREAM_ismi)

Master modlu cihazın I2C haberleşmesini başlatma komutudur.

İ2c_start();

İ2c_start(iletişim_1);

I2C_WRITE(veri) veya I2C_WRITE(STRAM_ismi,veri)

İ2c_write(0xA2);

İ2c_write(iletişim_1,0xA2);

I2C_READ() veya I2C_READ(ACK) veya I2C_READ(STREAM_ismi, ACK)

İ2c_read();

İ2c_read(0);

İ2c_read(iletisim_1,1);

<u>Örnek:</u>

İ2c_start();   // iletişimi başlat

İ2c_write(0xA0); // Slave cihazın adresi

İ2c_write(5); // Komut

İ2c_write(12); // Veri

İ2c_stop(); // iletişimi durdur

I2C Kesmesi:

#INT_SSP kesmesi kullanılır. Bu kesme I2C uçlarında okuma veya yazma tarzında bir aktivite olduğu zaman bu kesmeyi oluşturur.

Örnek: Harici bir cihaza(0x05 adresli) PIC ile I2C den veri yazma ve veri okuma:

```
#include <prog.h>
int1 hazirmi()
{
int1 hazir;
İ2c_start();
hazir=i2c_write(0x0A);
İ2c_stop();
return !hazir;
}
```

```
void yaz(byte komut, byte veri)
{
while(!hazirmi());
İ2c_start();
i2c_write(0x0A);
i2c_write(komut);
i2c_write(veri);
İ2c_stop();
delay_ms(20);
}
```

```
void oku(byte adres)
{
byte veri;
while(!hazirmi());
İ2c_start();
i2c_write(0x0A);
i2c_write(komut);
İ2c_start();
i2c_write(0x0B);
veri=i2c_read();
İ2c_stop();
return (veri);
}
```