OneWire Library

OneWire library provides routines for communication via OneWire bus, for example with DS1820 digital thermometer. This is a Master/Slave protocol, and all the cabling required is a single wire. Because of the hardware configuration it uses (single pullup and open collector drivers), it allows for the slaves even to get their power supply from that line.

Some basic characteristics of this protocol are:

- single master system,
- low cost,
- low transfer rates (up to 16 kbps),
- fairly long distances (up to 300 meters),
- small data transfer packages.

Each OneWire device also has a unique 64-bit registration number (8-bit device type, 48-bit serial number and 8-bit CRC), so multiple slaves can co-exist on the same bus.

Note that oscillator frequency *Fosc* needs to be at least 4MHz in order to use the routines with Dallas digital thermometers.

Library Routines

- Ow_Reset
- Ow_Read
- Ow_Write

Ow_Reset

Prototype	`sub function Ow_Reset(dim byref port as byte, pin as byte) as byte`
Returns	0 if DS1820 is present, and 1 if not present.
Description	Issues OneWire reset signal for DS1820. Parameters `port` and `pin` specify the location of DS1820.
Requires	Works with Dallas DS1820 temperature sensor only.
Example	To reset the DS1820 that is connected to the RA5 pin: `Ow_Reset(PORTA, 5)`

Ow_Read

Prototype	`sub function Ow_Read(dim byref port as byte, dim pin as byte) as byte`
Returns	Data read from an external device over the OneWire bus.
Description	Reads one byte of data via the OneWire bus.
Requires	Works with Dallas DS1820 temperature sensor only.
Example	`tmp = Ow_Read(PORTA, 5)`

Ow_Write

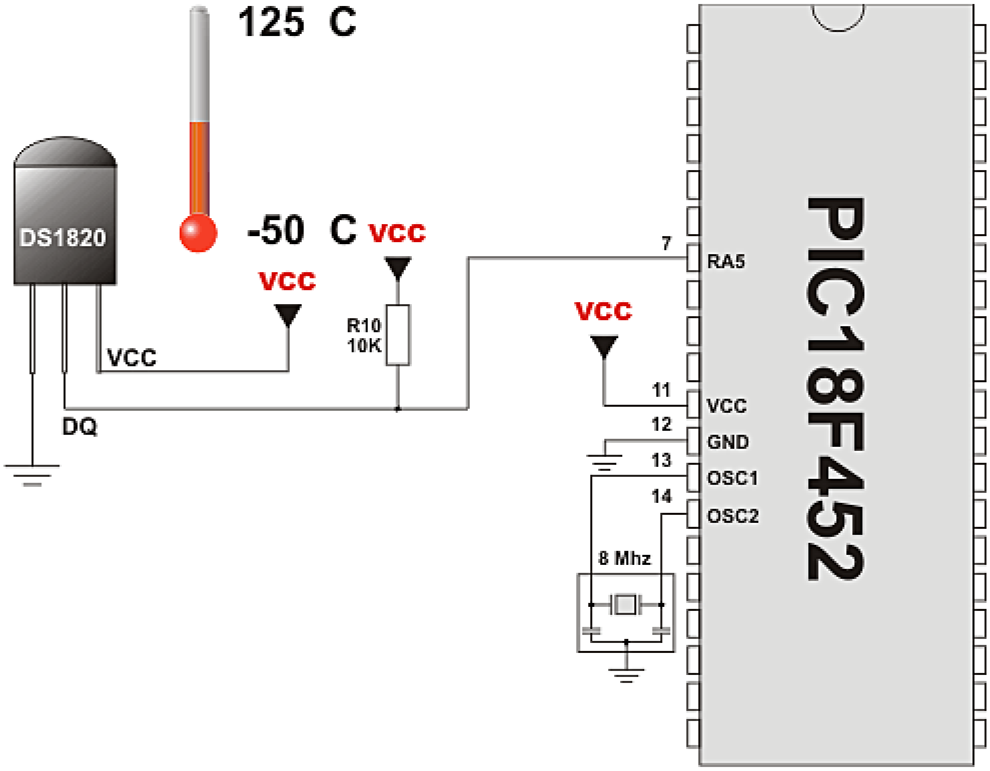
Prototype	`sub procedure Ow_Write(dim byref port as byte, dim pin, par as byte)`
Returns	Nothing.
Description	Writes one byte of data (argument `par`) via OneWire bus.
Requires	Works with Dallas DS1820 temperature sensor only.
Example	`Ow_Write(PORTA, 5, $CC)`

Library Example

The example reads the temperature from DS1820 sensor connected to RA5. Temperature value is continually displayed on LCD.

```
program onewire

dim i, j1, j2 as byte
    text      as string[6]
    tmp_sign  as byte

main:

   text   = "Temp:"
   adcon1 = 255                     ' configure RA5 pin as digital I/O
   PORTA  = 255                     ' initialize porte to 255
   PORTD  =   0                     ' initialize portb to 255
   TRISA  = 255                     ' designate porte as input
   TRISD  =   0                     ' designate portb as output
   Lcd_Init(PORTD)
   lcd_cmd(LCD_CURSOR_OFF)
   lcd_out(1, 1, text)
   lcd_chr(2, 12, 223)               ' 'degree' character
   lcd_chr(2, 13, "C")

   while TRUE
     ow_reset(PORTA, 5)              ' onewire reset signal
     ow_write(PORTA, 5, $CC)         ' issue command to DS1820
     ow_write(PORTA, 5, $44)         ' issue command to DS1820
     delay_us(120)
     i = ow_reset(PORTA, 5)
     ow_write(PORTA, 5, $CC)         ' issue command to DS1820
     ow_write(PORTA, 5, $BE)         ' issue command to DS1820

     j1 = ow_read(PORTA, 5)          ' get result
     j2 = ow_read(PORTA, 5)          ' get result (assuming the temperature is positive)

     if j2 = $FF then
       tmp_sign = "-"                ' temperature sign
       j1= j1 or $FF                 ' complement of two
       j1= j1 + $01
     else
       tmp_sign = "+"
     end if
     j2 = (j1 and $01) * 5           ' Get decimal value
     j1 = j1 >> 1                    ' Get temp value

     ByteToStr(j1, text)             ' whole number
     lcd_chr(2, 7, tmp_sign)
     lcd_chr(2, 8, text[1])
     lcd_chr(2, 9, text[2])
     lcd_chr(2, 10, 46)              ''.'

     ByteToStr(j2, text)             ' decimal
     lcd_chr(2, 11, text[2])

     Delay_ms(500)
   wend                              'endless loop
end.
```

HW Connection



Example of DS1820 on RA5