DALLAS SEMICONDUCTOR

Application Note 126
1-Wire® Communication through Software

www.dalsemi.com

INTRODUCTION

A host CPU can easily generate 1-Wire timing signals itself if a true bus master is not present. This is an example, written in C, of how to perform the communication. There are several system requirements for proper operation of the code:

1. The communication port is bi-directional, its output is open drain, and there is a weak pull-up on the line. This is a requirement of any 1-Wire bus.
2. The system is capable of generating an accurate and repeatable 1us delay.

This code relies on the two C functions "outp" and "inp" to write and read bytes of data to port locations. They are located in the <conio.h> standard library. Usage:

int outp(unsigned port, int byte);
int inp(unsigned port);

The constant PORTADDRESS in the code is defined as the location of the communication port. The code assumes bit 0 of this location is the 1-Wire bus.

The function "Waitx" in this example is a user-generated routine to wait a variable number of microseconds. This function will vary for each unique hardware platform running this code so it cannot possibly be described here. Usage:

//Pause for exactly x microseconds
void Waitx(int microseconds){}

FUNCTIONS

The four basic operations of a 1-Wire bus (Reset, Write 1, Write 0, and Read Bit) are listed next. They will be used to build more complex functions later.

```
// Generate a 1-Wire reset, return 1 if no
// presence detect was found, return 0 otherwise.

int Reset(void)
{
   int result;

   outp(PORTADDRESS,0x00); //Drives DQ low
   Waitx(480);
   outp(PORTADDRESS,0x01); //Releases the bus
   Waitx(120);

   //Sample and return the Presence Detect
   result = inp(PORTADDRESS) & 0x01;
   Waitx(360);

   return result;
}

// Generate a Write1.

void Write1(void)
{
   outp(PORTADDRESS,0x00); //Drives DQ low
   Waitx(1);
   outp(PORTADDRESS,0x01); //Releases the bus
   Waitx(59);
}

//Generate a Write0. I'm giving a 5us recovery
//time in case the rise time of your system is
//slower than 1us. This will not affect system
//performance.

void Write0(void)
{
   outp(PORTADDRESS,0x00); //Drives DQ low
   Waitx(55);
   outp(PORTADDRESS,0x01); //Releases the bus
   Waitx(5);
}

// Read 1 bit from the bus and return it

int Readx(void)
{
   int result;

   outp(PORTADDRESS,0x00); //Drives DQ low
   Waitx(1);
   outp(PORTADDRESS,0x01); //Releases the bus
   Waitx(14);

   //Sample after 15us
   result = inp(PORTADDRESS) & 0x01;  Waitx(45);

   return result;
}
```

This is all that is required to do bit-wise manipulation of the bus, however the above routines can be built upon to create byte-wise manipulator functions.

```
// Write data byte

void WriteByte(int Data)
{
   int loop;

   //Do 1 complete byte
   for(loop=0; loop<8; loop++)
   {
      //0x01,0x02,0x04,0x08,0x10,ect.
      if(Data & (0x01<<loop))
         Write1();
      else
         Write0();
   }
}

// Read data byte and return it

int ReadByte(void)
{
   int loop;
   int result=0;

   for(loop=0; loop<8; loop++)
   {
      result = result + (Readx<<loop);
   }

   return result;
}
```

These six functions plus the user’s “Waitx” function are all that are required for control of the 1-Wire bus at byte level. The following example shows how a user would bring those functions together by reading the ICA register of a DS2437.

```
//Read the Integrated Current Accumulator

int ReadICA(void)
{
   // Recall Page 1
   if(Reset()) return 0;  //No devices found

   WriteByte(0xCC);    //Skip ROM
   WriteByte(0xB8);    //Recall memory
   WriteByte(0x01);    //Page 1

   // Read Page 1
   Reset();
   WriteByte(0xCC);    //Skip ROM
   WriteByte(0xBE);    //Read memory
   WriteByte(0x01);    //Page 1
   ReadByte();         //Ignore first 4 bytes
   ReadByte();
   ReadByte();
   ReadByte();
   return ReadByte(); // The ICA register
}
```

If a software solution is unacceptable for the application, consult the DS1WM datasheet for an alternative solution.

[1] **DS1WM Datasheet**, Dallas Semiconductor, online at http://www.dalsemi.com/datasheets/pdfs/1wm.pdf