



Vogelaar Electronics

Dorpsstraat 90
3751 ES Bunschoten Netherlands
Telefoon +31 (0)33 2980727
Fax +31 (0)847 115096
E-mail info@vogelaar-electronics.com

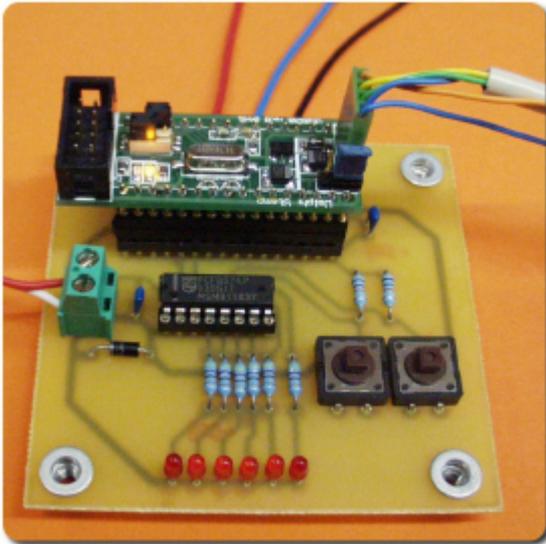
DS_AN_DevDrv.doc
18-09-2005

DelphiStamp VE08201

Device drivers
(Introduction)

by Vogelaar Electronics
Bunschoten, Netherlands
18 – September - 2005

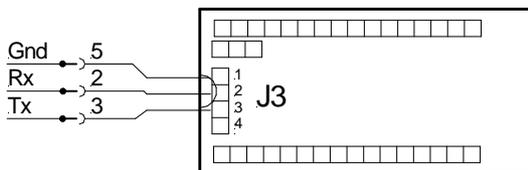
To assist in the development of the device driver the circuit as described above was built (part number VE09202).



The circuit is energized from an external +5 VDC power supply, Red (Gnd) and White (+5V) wire on screw terminal.

An oscilloscope is attached for monitoring the SDA and SCL lines (red, blue and black wires).

The RS232 connection cable required for uploading the firmware is connected to J3 on the DelphiStamp.



The I2C / TWI device driver.

The driver is written and compiled by using the PasAvr compiler. To achieve modularity the driver is put in a separate unit i.e. UDriverEx.pas. To start a directory named <path>\2 TWI\ is created. The template files from the directory <path>\1 PasAvr Start*. * are copied in <path>\2 TWI\.

The template files are:

UMain.pas

This file contains the "operating system". It starts with linker information stating : Link \$000,\$180. This means that user code will start from \$0000 growing towards \$1DFFF, and user RAM starts at \$180 growing towards \$10FF.

The dummy procedure StartVectors_Low contains a jump table with interrupts vectors including the power-on vector @Main.

The code block Main contains initialization of the stack pointer and timer #3 to 100 mSec intervals. Three external procedures are called that are located in the unit UControl.pas i.e.

Procedure ControlInit : called once to initialize the user application

Procedure ControlExe : called at T = 100 mSec. The actual user program

Procedure ControlIdle : called in when idle.

UDrivers.pas

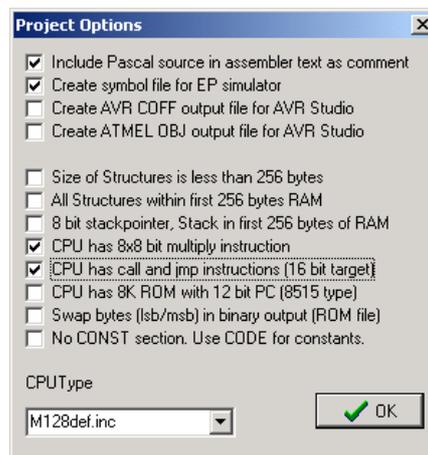
This is the template file containing the declarations for procedures and functions controlling standard hardware as located in the BIOS.

UControl.pas

This is the template file for an user application. This file is identical for Delphi and PasAvr applications.

1. AvrPas.exe

Start the PasAvr IDE



2. Project | New project

and save it as <path>\2 TWI\PTWI.prj

3. Project | Main file | <path>\2 TWI\Main.pas

4. Project | Add to project | <path>\2 TWI\Main.pas

5. File | Open | <path>\2 TWI\UDrivers.pas

6. File | Open | <path>\2 TWI\UControl.pas

7. File | New (non project file) | <path>\2 TWI\UDriversEx.pas

8. Edit UDriversEx.pas until it contains :

```
Unit UDriversEx;
(* Device driver unit demonstrating simple I2C / TWI using VE08201 DelphiDtamp.
   Used evaluation hardware : VE09202
   Provided by Vogelaar Electronics, Bunschoten Netherlands
   Rev 0.10 16-09-05 Initial release *)

(* ===== Interface ===== *)
Interface
```

```

Procedure TWI_Init;                                     // Set-Up of TWI controller
Procedure PCF8574_Wr (X : Byte); // Write output data X to I/O expander
Function  PCF8574_Rd : Byte;      // Read  input  data from I/O expander

(* ===== Implementation ===== *)
Implementation

Procedure TWI_Init;
(* Initialize TWI controller on AtMegal28 for 100 kHz use *)

Begin
  _PortD := $03;                                     // Enable pull-up on SCL and SDA
  _TWCR  := 0;                                       // Reset TWI
  _TWBR  := 35;                                     // TWI clock := 50 kHz
  _TWSR  := 1
End;

Procedure PCF8574_Wr (X : Byte);
(* Simple byte write to I/O extender with no status checking *)

Begin
  _TWCR := $A4;                                     // TWI Start
  Repeat Until _TWCR And $80 <> 0;                 // Wait until done
  _TWDR := $40;                                     // PCF8574 Addr:0 Write
  _TWCR := $84;                                     // TWI Write
  Repeat Until _TWCR And $80 <> 0;                 // Wait until done
  _TWDR := X;                                       // Set Data
  _TWCR := $84;                                     // TWI Write
  Repeat Until _TWCR And $80 <> 0;                 // Wait until done
  _TWCR := $94;                                     // TWI Stop
End;

Function PCF8574_Rd : Byte;
(* Simple byte read to I/O extender with no status checking *)

Begin
  _TWCR := $A4;                                     // TWI Start
  Repeat Until _TWCR And $80 <> 0;                 // Wait until done
  _TWDR := $41;                                     // PCF8574 Addr:0 Read
  _TWCR := $84;                                     // TWI Write
  Repeat Until _TWCR And $80 <> 0;                 // Wait until done
  _TWCR := $84;                                     // TWI Read + NAK
  Repeat Until _TWCR And $80 <> 0;                 // Wait until done
  Result := _TWDR;                                  // Get Data
  _TWCR := $94;                                     // TWI Stop
End;

(* ===== End ===== *)

End.

```

9. Modify UControl.pas into :

```
Unit UControl;
(* Control unit demonstrating simple I2C / TWI using DelphiDstamp.
   Used evaluation hardware : VE09202
   Provided by Vogelaar Electronics, Bunschoten Netherlands
   Rev 0.10 16-09-05 Initial release *)

(* ===== Interface ===== *)
Interface

{*} Uses    UDrivers, SysUtils; {*}           // For Delphi
{!*} Uses   UDrivers, UDriversEx, IntLib;  !*} // For PasAvr

Procedure ControlInit;           // Called at start of program
Procedure ControlExe;           // Called at T = 100 mSec
Procedure ControlIdle;          // Called during spare time

(* ===== Implementation ===== *)
Implementation

Var    X      : Byte;           // Led pattern on VE09202

Procedure ControlInit;
(* Called once at start of program *)

Begin
    TWI_Init
End;

Procedure ControlExe;
(* Called at T = 100 mSec. Busy time 800 uSec *)

Begin
    X := PCF8574_RD;
    X := X Shr 2 Or $C0;           // Switch wires as input
    PCF8574_WR (X);
    Led2Toggle                     // 5 Hz on DelphiStamp
End;

Procedure ControlIdle;
(* Called during spare time *)

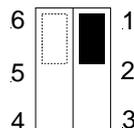
Begin
End;

(* ===== End ===== *)

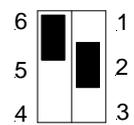
End.
```

10. Click on tab UDrivers.pas
11. Compile | Current editor file
12. Click on tab UDriversEx.pas
13. Compile | Current editor file
14. Click on tab UControl.pas
15. Compile | Current editor file
16. Compile | Main project file
17. Close PasAvr

18. Reset Delphi stamp by setting jumpers

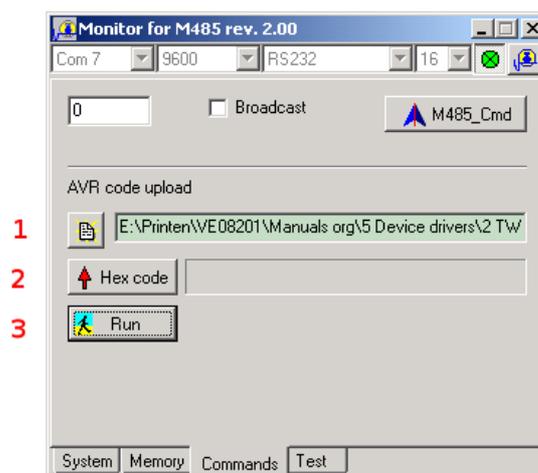
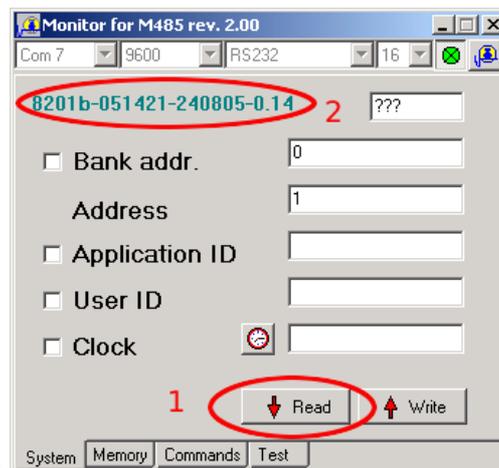


19. Start M485 server by setting jumpers



Now Led1 flashes at 0.5 Hz

20. Start Mon485.exe and test communication



21. Upload it and see it working :

Remarks : the TWI device driver can be improved by 1) testing result states, 2) using interrupts and a state machine and 3) using assembly language in time critical sections.