

# Edupic User's Guide

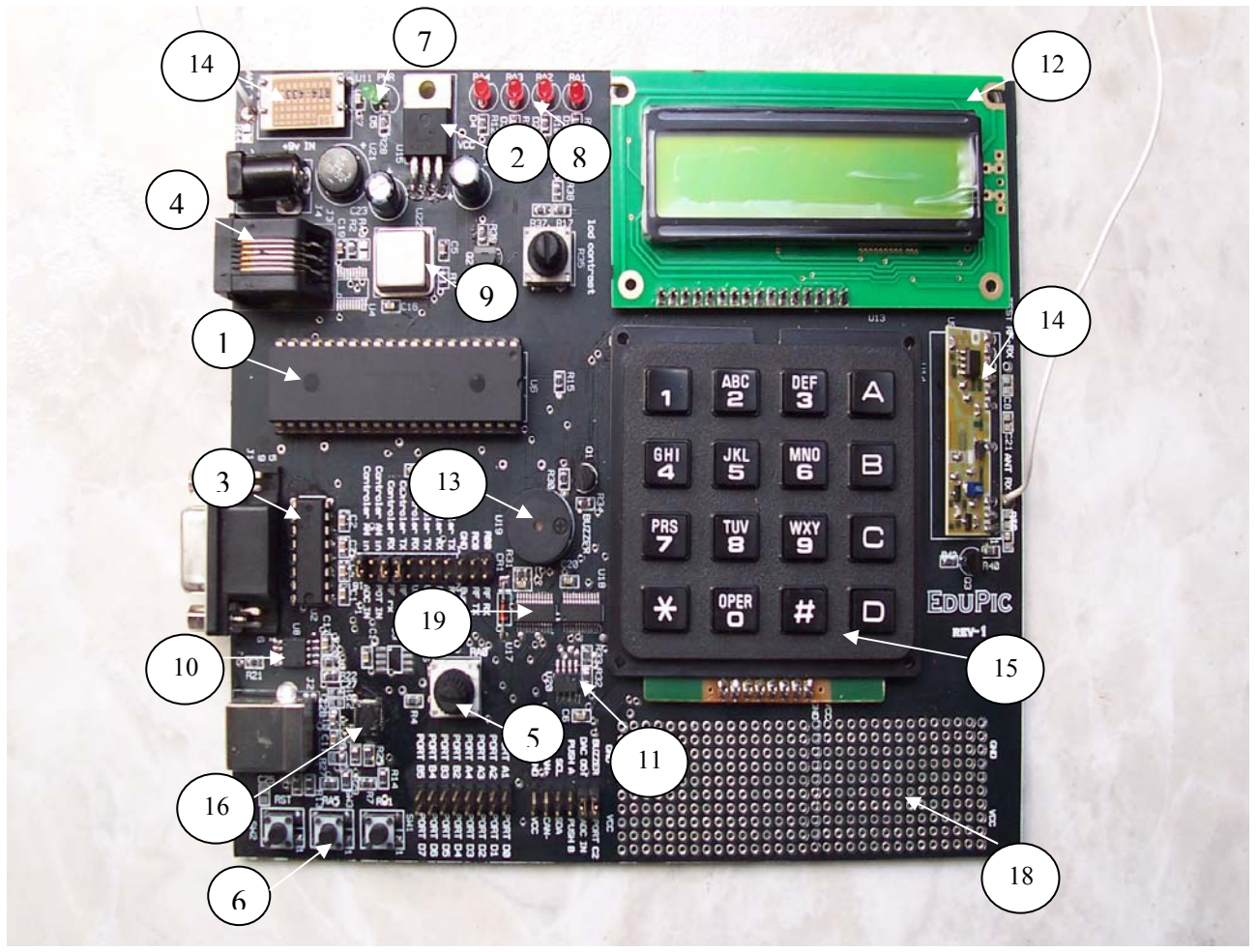
<b>Table of Contents:</b>	<b>Page</b>
1. Edupic Demonstration Board.....	2
2. Sample Device.....	4
3. Tutorial Program Operation.....	4
4. Hardware Details.....	7
5. Included Drivers.....	8

## **Edupic Demonstration Board**

The Edupic demonstration board has the following hardware features:

1. 40-pin Dip socket.(pic16f877 included)
2. On board 5v regulator for direct input 9v/100ma AC/DC adapter.
3. RS-232 associated hardware for direct connection to an RS-232 interface.
4. In-Circuit Debugger (ICD) connector.
5. 5 (k $\Omega$ ) pot connected to analog input.
6. Three push button switches, two for external stimulus and one for Reset.
7. Power-on indicator Led.
8. Four red LEDs connected to PORTA.
9. 16 Mhz crystal oscillator.
10. 32K x 8 Serial EEPROM.
11. Serial DAC with 8 bit resolution.
12. LCD Display.
13. Piezo buzzer.
14. AM Receiver and Transmitter prototypes for establishing wireless connection.
15. 4 x 4 Keyboard connected to PORTB(4bits) and PORTD(4bits).
16. USB associated hardware for direct connection to USB interface.
17. CAN-BUS interface (optional).
18. Prototype area for user connections.
19. Two 8 bit Q-Switches for electrical isolating and external connection of user area to the MCU.

# Edupic Hardware



## **Sample Device:**

There is FLASH device included (pic16f877).The user can change the device to other PIC16 or PIC18 40-pin DIP device.

## **Tutorial Program Operation:**

The tutorial program is made of nine subroutines, witch are individually displayed on the LCD.

### **1. Running leds mode.**

\*LEDS\* message displayed on the LCD first line.

The leds start running left. Pressing 1 on the keyboard will run the leds to the left, Pressing 2 will run the leds to the Right. Press D on the keyboard for moving to the next menu.

### **2. Buttons mode.**

\*BUTTONS\* message displayed on the LCD first line.

Pressing SW1 will turn on the D1 led until SW1 is released. Pressing SW3 will turn on the D2 led until SW3 is released. Press D for next menu.

### **3. ADC mode.**

\*A2D\* message displayed on the LCD first line.

This mode uses ADC to measure the voltage from the pot and display the voltage ( 0.00v to 5.00v) on the LCD second line. The pot is connected to analog input AN0.The voltage is updated continuously on the LCD display. The user must connect the jumper P1(13-14) Analog\_Pic\_In to Pot\_In. Press D for next menu.

### **4. DAC mode.**

\*D2A\* message displayed on the LCD first line.

In this mode the user can send the wanted voltage value via Serial I2C Bus to the DAC and than check the actual voltage from the DAC output by connecting it to AN0 analog input of the MCU. The user must connect the jumper P1(15-16) Analog\_Pic\_In to ADC\_In.

The first digit will blink D=0.00v (The maximum voltage is 2.50v so the user must enter the voltage below this value). Enter number from the keyboard (0-2) and than Press # for next digit. The second digit will blink D=0.00v.Enter number and Press # for next digit. The third digit will blink D=0.00v. Enter number and Press #. The first digit will blink again. At this moment the wanted voltage will be sent to the DAC and the user should see the actual voltage on the LCD second line (for example D=1.25v A=1.25v). Press D for next menu.

## 5. **Buzzer mode.**

\*Buzzer\* message displayed on LCD first line.

In this mode the buzzer connected to RC2 pin of the MCU. By activating the CCP1 PWM mode, MCU supplying rectangular wave to pin RC2. The user can change the Period and The Duty Cycle of the wave. The buzzer is activated until the user moves to the next mode. The user can change the PR2 register(0-255) and the D.C. (0-99 %) by entering this values from the keyboard. At the start first digit will blink (LCD display second line -> PR2=000 D.C.=00%). The user should enter the wanted value for the digit and Press # (like in mode 4) .When PR2 and Duty Cycle is updated the sound from the buzzer will change. The user must connect the jumper Buzzer to PORT C2. Press D for next menu.

## 6. **RS232 mode.**

\*RS232\* message displayed on LCD first line.

In this mode the user can transmit and receive data via RS232 interface. The user must connect the jumpers P1(1-2) Controller\_TX to RS232\_TX and P1(3-4) Controller\_RX to RS232\_RX. The user has to options in the main menu. Press 1 to Receive or Press 2 to Transmit (LCD display second line RCV=1 TX=2).

In Receive mode the user can see on the LCD the last character that were received from the Computer (or other MPU with RS232 interface) for example GETCH=A where A is the last character sent via RS232 interface. Press D to Exit to the main menu.

In Transmit mode the user should send the character to the Computer by Pressing any key except D on the keyboard. The character will blink (PUTCH=A). Every time the user presses on the keyboard another character will be sent via RS232 interface. Press D to Exit to the main menu. Pressing on D while the user is in the main menu will go over the next mode.

## 7. **USB mode.**

\*USB\* message displayed on LCD first line.

In this mode the user can transmit and receive data via USB interface. The user must connect the jumpers P1(5-6) Controller\_TX to USB\_TX and P1(7-8) Controller\_RX to USB\_RX. The main menu is the same as in mode 6.

## 8. **External EEPROM mode.**

\*EEPROM\* message displayed on LCD first line.

The user can send to and receive from the EEPROM 8 bit data via I2C interface using this mode. There are two options in the main menu. Press 1 to Store the data in the EEPROM or Press 2 to read the data from EEPROM (LCD display second line -> 1-Store 2-Read).

**In Store mode** the user can enter number or character from the keyboard and store it in the specific address (0-999). For example if the user wants to Store A in address 2 than the values on the LCD should be like this CHAR=A ADD=002. Press D to Exit to the main menu.

**In Read mode** the user can read number or character from the specific address (0-999) by entering it from the keyboard. For example if character B was stored in address 5 than if the user will check this address the message on the LCD will be ADD=005 CHAR=B. Press D to Exit to the main menu.

Pressing on D while the user is in the main menu will go over the next mode.

## 9. **RF mode.**

\*RF\* message displayed on LCD first line.

In this mode the user can send to and receive data from RF blokes ( AM Transmitter and Receiver).The RF blokes data in and data out connected to RS232 RX and TX enabling wireless RS232 communication. The user must connect the jumpers P1(9-10) Controller\_TX to RF\_TX and P1(11-12) Controller\_RX to RF\_RX .The user can transmit number or character by Pressing any key except D on the keyboard and see the last value that received on the LCD second line (PUTCH=A GETCH=A). Since RF mode is the last one Pressing on D will go over the first mode (\*LEDS\*).

## Hardware Details:

1. **Processor Socket:**  
40-pin socket is provided to connect any PIC16 or PIC18 40-pin devices.
2. **Display:**  
Four red LEDs are connected to PORTA. PORTA pins are set low to light the LEDs.  
Another red LED is provided to determine whether there is power to the Edupic. Power on (LED on) Power off (LED off).
3. **Power Supply:**  
A 9v ,100ma unregulated AC or DC supply can be plugged into J4.  
MPLAB ICD 2 users may use the ICD to power the target board to 5v, up to 200ma, if MPLAB ICD 2 is connected to the PC with serial cable.
4. **RS232 Serial Port:**  
An RS232 level shifting IC has been provided with the necessary hardware to support connection of an RS232 host through the DB9 connector (J1).  
The port can be connected to PC using strait-through cable.  
The PIC16/18 RX and TX pins are tied to RX and TX pins of MAX232 through jumper P1(1-2 and 3-4).
5. **USB Port:**  
A RS232 to USB IC (CP2101) has been provided with the necessary hardware to support connection of a USB host through USB connector (J2).  
The PIC16/18 RX and TX pins are tied to RXD and TXD pins of CP2101 through jumper P1(5-6 and 7-8).
6. **CAN BUS:**  
A SPI (Serial 3 wire connection) to CAN BUS IC's (MCP2510 and MCP2551) has been provided with the necessary hardware to support connection of a CAN BUS.(optional)
7. **AM Transmitter and Receiver:**  
The Transmitter and Receiver RF blokes included on the Edupic board to enable wireless connection.  
The Transmitter data in connected to TX of the MCU through jumper P1(9-10) and the Receiver data out connected to RX of the MCU through jumper P1(11-12).When the RF blokes connected this way the user can establish RS232 wireless connection between to boards. (up to 50m distance).
8. **Switches:**  
S1-Active-low push button connected to RB1.  
S2-Active-low push button connected to MCLR to hard reset the processor.  
S3-Active-low push button connected to RA5.
9. **Oscillator:**  
Crystal Resonator of 16Mhz connected to OSC1 pin 13 of the MCU.

#### 10. **Analog Inputs:**

The 5k $\Omega$  pot or DAC analog output can be connected to analog input AN0 of the MCU through jumper P1(13-14) to connect the pot or through jumper P1(15-16) to connect the DAC.

Analog inputs connected to AN0 through serial resistor 422 $\Omega$ .

The pot can be adjusted from Vdd to Gnd (0-5v) and the DAC maximum voltage is Vref (2.5v).

#### 11. **ICD connector:**

The MPLAB ICD 2 can be connected for low-cost debugging through connector J3. The ICD utilizes RB6 and RB7 pins of the MCU for in-circuit-debugging.

#### 12. **Digital to Analog converter:**

Serial 8 bit DAC is included. The DAC TC1320 connected via I2C 2 wire bus (RC3 and RC4 pins of MCU). DAC has an address - 1001000b. The user must include d2a.c driver to use the DAC.

#### 13. **Serial EEPROM:**

A 24LC256 256kbits (32kbytes) serial eeprom connected to the MCU via I2C bus. EEPROM has an address – 1010100b. The user must include eeprom.c driver to use the external eeprom.

#### 14. **LCD:**

An LCD display with two lines , 16 characters each, connected to the MCU. There are three control lines (RE3-RE0) and four data lines (RD3-RD0). The R35 pot is installed to adjust the contrast of the LCD. The user must include lcd.c driver in his project to use the LCD.

#### 15. **Keyboard:**

The 4 X4 (16 characters) keyboard connected to the MCU to give the user an option for input data (numbers and characters). The four rows of the keyboard are connected to PORTD (RD7-RD4) and four columns are connected to PORTB (RB5-RB2). The user must include the kbd.c driver in his project to use the keyboard.

#### 16. **Buzzer:**

The buzzer connected to RC2 pin of the MCU. The buzzer included on the PCB to illustrate the concept of using MCU PWM (pulse width modulation) mode.

### **Included Drivers:**

1. Driver for using LCD module (lcd.c)
2. Driver for using Keyboard (kbd.c)
3. Driver for using External EEPROM (eeprom.c)
4. Driver for using Serial Digital to Analog Converter (d2a.c)