This report will discuss the design and development of a burglar alarm as a part of the Electronics and Embedded Hardware module. The module was constructed on a foundation of groupwork and therefore the report will reflect the individual contributions of two authors. For the purpose of this assignment, the contributors will be referred to as Author 1 and Author 2. An analysis of the technologies used to create a fully functioning burglar alarm using the QL-200 Development Board, will be presented, and will incorporate an examination of the driver code employed to control the devices. A critical evaluation will highlight the individual contributions and challenges faced. A reflection of these challenges and resolutions will then be discussed.

The overall project aim was to utilise a QL-200 development board and PIC16F877A microcontroller to create a burglar alarm which allowed for user input. The device monitors four separate security zones which can be activated or deactivated manually. Zone four contains a temperature based sensor which activates the alarm if the temperature rises above a threshold for a predetermined amount of time. The temperature threshold, activation time and alert time can be set using the 4x4 matrix buttons when the devices is first powered. An LCD panel would be employed to display information pertinent to the user such as the temperature of Zone 0, the current date and time and the activation status of each zone.

The DS18B20 temperature sensor was used in the development of the alarm to provide an accurate temperature measurement to one decimal place in degrees Celsius. This device utilises a three-pin setup, two for power and ground and one for communication, this design allows data input and output to be performed utilising only the communication pin.

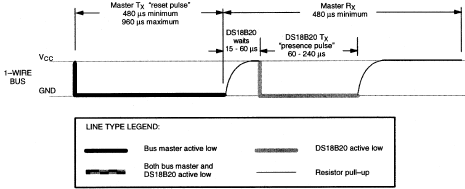
The initialisation process of the temperature sensor establishes communication between the device and the bus master, allowing it to send commands to the sensor and interpret received data. This process requires a reset pulse to be transmitted by the bus master, followed by a presence pulse from the temperature sensor notifying the bus master that the sensor is present and ready to accept commands. The timing of this process is vital and must be within a specific range otherwise, this could output incorrect result. This is show in Figure 1.

Figure 1

*Initialisation Procedure for DS18b20 Temperature Sensor*

Figure 1 describes for the reset pulse to be successful, the DQ pin must be pulled low for a maximum of 960 microseconds and then pulled high. The sensor must wait for a maximum of 60 microseconds, this is to transition between the transmission of the reset pulse from the bus master to the presence pulse emitted from the sensor and interpreted by the bus master.

Rom Commands are 8-bit sequences written in hexadecimal that are transmitted to the temperature sensor providing a number of useful processes. The bits are sent individually to the sensor and a bit-shift operation and if statement is performed to determine the bits individual value. The Skip Rom command (CC in hex) was issued to provide an efficient means for the bus master to access the memory functions, this process must only be attempted if there is a single temperature sensor in use as this could cause a data collision.

A temperature conversion of the sensor data is then required to output the correct binary code. This command – denoted as 44 in hexadecimal – stores 16 bits of data into the devices RAM known as the scratchpad in a two’s compliment format. An example of this is shown in Figure 2.

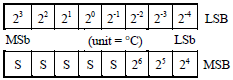


Figure 2

*The output after transmitting a temperature conversion command.*

This format presents a problem however, the lower four bits of the LSB contain information required for the decimal point and the four highest bits of LSB and lowest four bits of MSB contain the data pertinent to the integer segment of the temperature. To resolve this, a read command is issued which retrieves the bits in the scratchpad, utilising a bit-shift operation the bits are read individually to determine whether the value is 0 or 1. Two variables should then be initialised to store the LSB and MSB.

Another bit-shift operation is required to gather the integer and decimal bits inside LSB and MSB and store the output into two separate variables. The output of this bit manipulation is shown in Figure 3.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| S | S | S | S | 2-1 | 2‑2 | 2-3 | 2-4 |

Integer Byte

Decimal Byte

Figure 3

*The output after performing bit-shift operation on LSB and MSB to find the integer and decimal values.*

Displaying the result is the final part of the temperature process, this requires the use of a LCD module. The details of this module are explained later in the report.

As depicted in the coursework specification, the burglar alarm should contain specific time constraints. These constraints include the alarm activating and deactivating at specific times in the day, the amount of time the temperature has risen above the threshold for and the duration of the alarm siren. To allow for these constraints, the DS1302 timekeeping chip was implemented. This eight-pin integrated circuit provides the time in either a 12 or 24-hour format and the day, date, month and year. Communication with this chip is achieved using the serial clock (SCLK), Data Input/Output (IO) and Reset (RST) pins.

Initialisation of the timekeeping chip requires SCLK and RST to be pulled low, this is to allow RST to then be pulled high allowing for data transmission to proceed. This oscillation is due to the requirement of SCLK remaining low when RST is being pulled high. Before any write operation is allowed, the write protect bit located at the control resister address must be zero, this manipulation is vital as the bit prevents any write operation to the clock or RAM if the value is 1. Accessing the control register – denoted as 8E in hexadecimal - requires each bit to be individually read utilising a bit shift operation. The write protect bit located at bit seven of the control register can then be set to zero. The other seven bits are forced to 0. RST is then pulled low.

Setting the time is achieved by using a clock burst write command denoted as BE in hex. This command allows the clock and calendar registers to be consecutively written to including the control register.

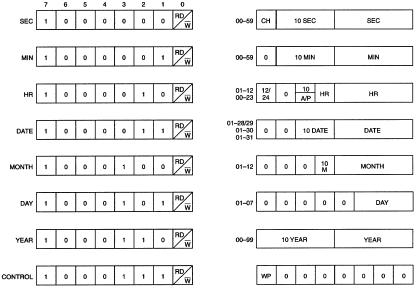
Performing a clock burst read command denoted as BF in hexadecimal transfers the current time into a secondary set of registers. It is the second register set that is read for the timing information.

Figure 4

*The clock and calendar register and the location of the units and tens column*

Values such as seconds are stored in separate sections of the respective byte, the units’ column is stored in bits 0, 1, 2, 3 while the tens column is stored in bits 4, 5, 6 as shown in Figure 4. Displaying the time therefore requires the time to be split into units and tens by performing a bit-shift operation on each of the timing register bytes.