

SECURE DIGITAL ACCESS SYSTEM USING IBUTTON

Access control forms a vital link in a security chain. Here we describe a secure digital access system using iButton that allows only authorised persons to access a restricted area.

The iButton is used here as a key to the access control system. Its unique identification (ID) number is used for authorisation. On detection of an authorised iButton, the system allows access. Thereafter, an automated lock key locks the system again. The system is permanently halted after five repeated false attempts. A service control unit built around an AVR microcontroller is interfaced to the iButton with 1-wire protocol for authentication of user validation of data.

iButton DS1990A

Here we have used the iButton DS1990A from Dallas Semiconductor (MAXIM).

Its block diagram is shown in Fig.1.

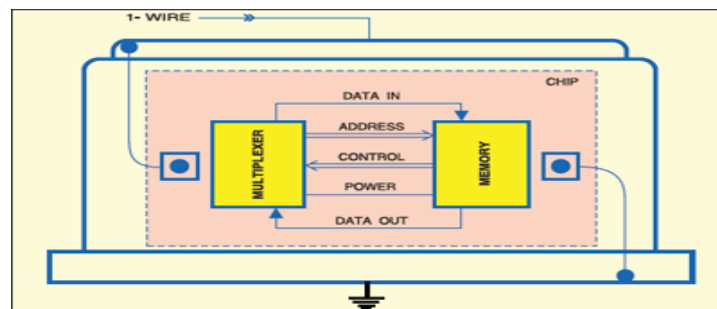


Fig.1: Block diagram of iButton

An iButton is a chip housed in a stainless-steel enclosure (refer Fig.2). The electrical interface is reduced to the absolute minimum, i.e., a single data line plus a ground reference. The energy needed for operation is taken from the data line. The DS1990A serial number iButton is a rugged data carrier that acts as an electronic registration number for automatic identification. It contains a unique ROM code that is 64-bit long as shown in Fig.3. The first eight bits are a 1-wire family code. The next 48 bits are a unique serial number. The last eight bits are a cyclic redundancy check (CRC) of the first 56 bits.



Fig.2: A typical iButton chip

Data is transferred serially via the 1-wire protocol, which requires only a single data lead and a ground return. The iButton DS1990A provides the additional 1-wire protocol capability that allows the search ROM command to be interpreted by the DS1990A.

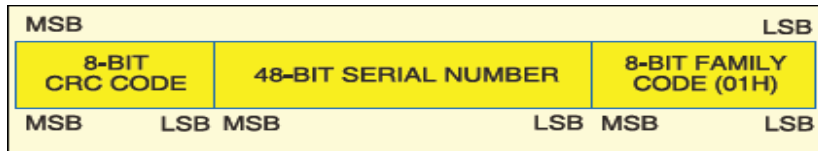


Fig.3: 64-bit lasered Rom

Circuit description

Fig.4 shows the circuit of the secure digital access system using iButton. The circuit is built around an ATmega16 microcontroller.

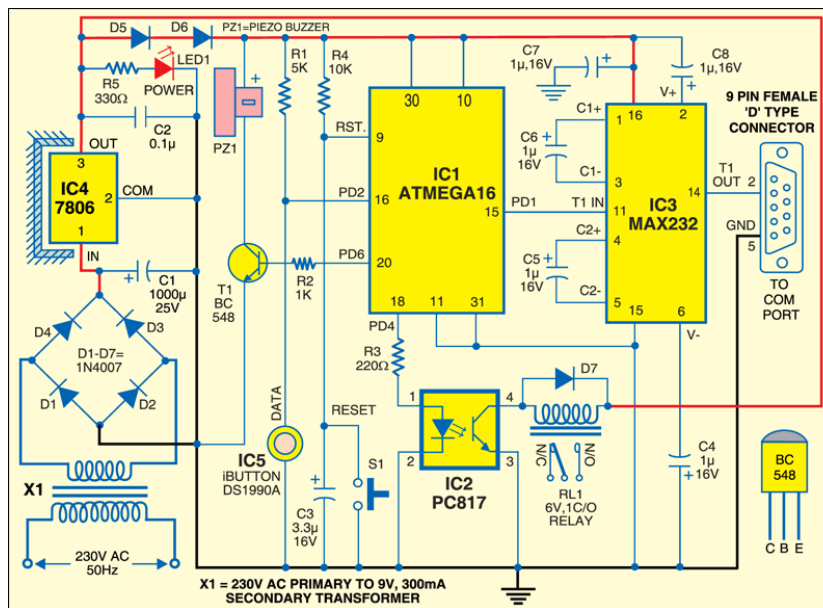


Fig.4: Circuit of the secure digital access system using iButton

PARTS LIST	
<i>Semiconductors:</i>	
IC1	- ATmega16 microcontroller
IC2	- PC817 optocoupler
IC3	- MAX232 RS-232 driver
IC4	- 7806, 6V regulator
IC5	- DS1990A iButton
T1	- BC548 npn transistor
D1-D7	- 1N4007 rectifier diode
LED1	- 5mm LED
<i>Resistors (all ¼-watt, ±5% carbon):</i>	
R1	- 5-kilo-ohm
R2	- 1-kilo-ohm
R3	- 220-ohm
R4	- 10-kilo-ohm
R5	- 330-ohm
<i>Capacitors:</i>	
C1	- 1000µF, 25V electrolytic
C2	- 0.1µF ceramic disk
C3	- 3.3µF, 16V electrolytic
C4-C8	- 1µF, 16V electrolytic
<i>Miscellaneous:</i>	
X1	- 230V AC primary to 9V, 300mA secondary transformer
RL1	- 6V, 1C/O relay
PZ1	- Piezobuzzer
S1	- Push-to-on tactile switch 9-pin D-type female connector

The ATmega16 is an 8-bit microcontroller based on the AVR enhanced RISC architecture that executes powerful instructions in a single clock cycle. It has 16kB in-system programmable flash program memory with read while- write capabilities, 512 bytes of EEPROM, 1kB SRAM, 32 general purpose input/output (I/O) lines, 32 general-purpose working registers, three flexible timers/counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented two-wire serial interface, a programmable watchdog timer with internal oscillator, an SPI serial port and six software-selectable power-saving modes.

Piezobuzzer PZ1 is used as an audible indicator for true, fake, random touches and system halt. It is controlled from port pin PD6 of the microcontroller with the help of transistor T1. The iButton socket is connected to port pin PD2. Pull-up resistor R1 is used as required from the 1-wire protocol. Port pin PD4 controls the relay operation through optocoupler PC817 (IC2). The door locking mechanism, say, for a slide door, is connected to the contacts of relay RL1, which closes after some time automatically.

Whenever someone attempts to gain access by touching the iButton (IC5) using his own iButton, the firmware inside the AVR reads its unique ID and matches with the ID in the firmware. If the ID matches, port pin PD4 goes high, the internal LED of the optocoupler (IC2) glows and relay RL1 energises for the predefined time. Simultaneously, the buzzer sounds to indicate grant of access.

Thereafter, relay RL1 de-energises. The buzzer gets the PWM signal to produce sound from port pin PD6 of the microcontroller. The 1-wire communication is done through interrupt port pin PD2 (INT0), so the AVR does not have to poll the pin but respond when any change in signal is detected on the interrupt pin. Switch S1 is used for manual reset.

Port pin PD1 of the microcontroller is used to interface with the hyper terminal of the PC through RS-N 232 interface MAX232 IC (IC3) for iButton verification and checking. The microcontroller provides a transmit channel for serial data transfer.

Transmit data pin (TXD) is specified at port pin PD1. The microcontroller is connected to T1 IN (pin 11) of MAX232. T1 OUT (pin 14) of IC3 is connected to pin 2 of the Comport connector. The signals provided on these pins are TTL-level and must be boosted and inverted through a MAX232 converter to comply with the RS-232 standard.

The MAX232 has two internal charge pumps that convert +5V into $\pm 10V$ (unloaded) for RS-232 driver operation. The first converter uses capacitor C6 to double the +5V input to +10V on capacitor C8 at pin 2. The second converter uses capacitor C5 to invert +10V to -10V on capacitor C4 at pin 6.

The power supply for this circuit is derived from 230V, 50Hz AC mains.

Transformer X1 steps down 230V, 50Hz AC mains to deliver a secondary output of 9V, 300 mA. The transformer output is rectified by a full-wave rectifier comprising diodes D1 through D4, filtered by capacitor C1 and regulated by IC 7806 (IC4). Capacitor C2 bypasses the ripples present in the regulated supply.

LED1 acts as the power indicator and R5 limits the current through

LED1. Construction and testing An actual-size, single-side PCB for the secure digital access system is shown in Fig. 5 (View as PDF) and its component layout in Fig. 6 (View as PDF). Assemble the circuit on a PCB as it minimises time and assembly errors. Carefully assemble the components and double-check for any overlooked error. Connect the assembled circuit to the COM port of the computer.

The iButton access code and message is transferred to the PC through the COM port using the Hyper Terminal program.

Software

The software for this project is given at the end of this article. It is written in 'Basic' language and compiled using Bascom-AVR compiler. The source program is well commented and easy to understand.

Burn the finally obtained Intel hex code file into the AVR's flash memory using a suitable programmer. The microcontroller uses an 8MHz internally generated clock. To activate, program fuse bytes as follows:

Fuse low byte = D4

Fuse high byte = 99

The source program is developed quickly with the help of the BASCOMAVR library function. The `1wrest`, `1wread` and `1wwrite` functions are used for checking the presence of 1-wire device, reading of the unique ID of iButton and writing to the 1-wire device, respectively. `Sound` function is used for generating the beep sound from the buzzer. The iButton is connected to hardware interrupt pin INT0 so that any change on the pin can be detected and further processing for 1-wire done. The program execution starts by initialising the input/output ports, the interrupt pin and its level of detection. Global interrupts are enabled and the interrupt service routine is ready to be executed when any interrupt is received.

As soon as the normal program execution is interrupted, an interrupt is issued and the control of execution enters the interrupt service routine, where iButton is given reset command first and then the ROM command to fetch the unique ID of the iButton. This fetched ID is compared with the unique ID programmed in the firmware. If there is a mismatch between the two IDs, a fake parameter counter is incremented and the relay connected to port PD4 de-energises. If the two IDs match, the relay connected to port PD4 energises and simultaneously the buzzer connected to port PD6 sounds. If the unique ID is wrong sequentially five times, the system enters the major warning state where the buzzer sounds continuously and it can only be stopped by resetting the system

Source: <http://todayscircuits.blogspot.com/2013/06/secure-digital-access-system-using.html#.VUCUsdKqpBc>