Knowledge Exchange

MIAC Electronic Door Lock

by Sean King, June 2010

Abstract

Many people don't realise the MIACs potential for real world, everyday applications, in this article we are shown how the versatile MIAC device coupled with the powerful programming language of Flowcode can be used to quickly and easily develop an electronic door lock with password security, and all running at 12V!

Requirements

Software:

- Professional or MIAC licence of Flowcode v3 or v4 for PIC.

Hardware:

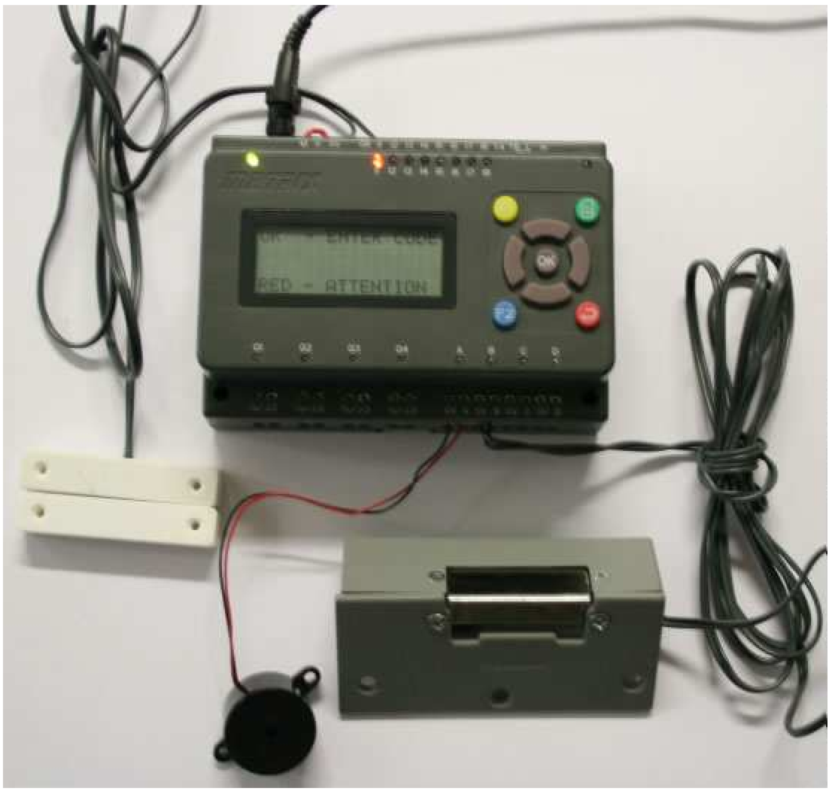
- MIAC device

Introduction

This article shows how a MIAC can be used to control and monitor a door entry system.

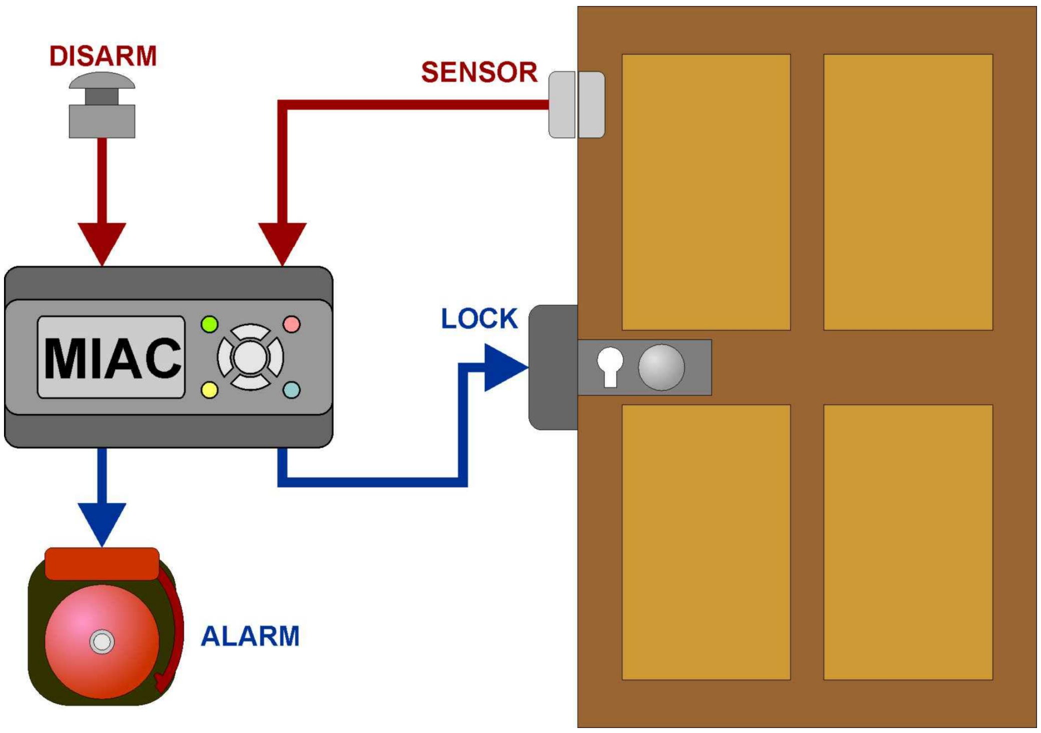
The 12V operation of the MIAC makes it compatible with wide range of domestic and industrial alarm/ security components. A low cost electronically released lock mechanism, and door contact reed switch from a standard alarm system, can be connected directly to a MIAC to provide access control.

The system consists of:
• A MIAC
• A solenoid operated lock release mechanism
• A door sensor
• An audible indicator
• A disarm button (optional)



The picture to the right shows the sample setup used when creating this article. All of the required listed parts can be seen.

Application
A system of this type can be configured in a number of different ways. In this instance the MIAC will be positioned close to the locked door to provide the display and keypad at the access point.



The door can be unlocked automatically by entering the correct key sequence on the keypad. Alternatively, the sounder can be used to attract attention by providing a 'door bell' option, allowing the door to be unlocked manually from the secure side.

The audible indicator produces several different sounds to indicate various conditions:
• Attention required (door bell)
• Lock released
• Door open
• Alarm (door left open, forced open)
• Disarm button (temporary override)

Program loop
The program is written as a single loop with macro calls to handle the individual system elements (door sensor, keypad, lock, sounder, display).

None of the macros are allowed to retain control of the system, so the whole program cycles continuously. This is important to allow all conditions to be continually monitored and reacted to.

Operating states
The current state of each system element is stored in global variables. These can be used by other elements to calculate their required states and actions.

Example:
If the door lock state is 'Locked', but the door sensor state is 'Open', the sounder should go to its 'Alarm' state.

In some cases it is preferable to detect changes of states to initiate an action. Where this is necessary, second state variable, prefixed OLD_, is added to retain the value of the associated state on the previous cycle of the loop. General or specific changes can be detected by comparing two copies of the state variable.

Timers
Some of the system operations require relatively long time delays (several seconds). Generating these delay functions, without suspending operation of other functions, is achieved by implementing them as loop counters. These rely on the continuous cycling of the program to provide the main time base. Absolute timing accuracy is not important in this application, so a simple delay in each loop cycle is used to guarantee a minimum period. Greater accuracy could be achieved by using a timer interrupt to trigger the execution of each loop.

If a delay is required, a value is placed in a variable being used by the appropriate delay timer. The variable is tested on each cycle of the program:
• if the value is not zero, it is decremented
• if the value is zero, the delay time has elapsed

System states
The main requirements of the system are:
• momentarily release the door lock when specific conditions are met
• confirm that the door is not opened when it is expected to be closed

The individual system components can be in the following states:

DOOR SENSOR -
Closed, Open

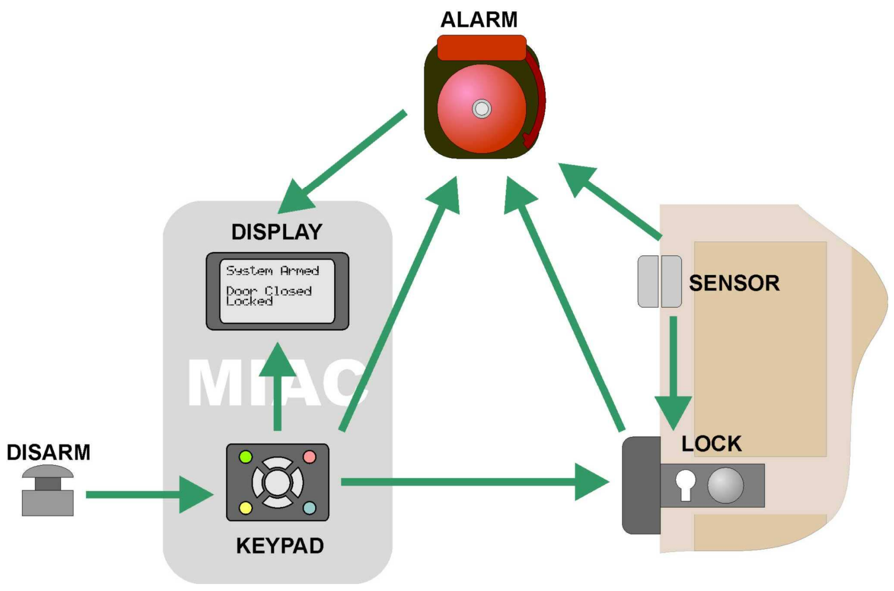
LOCK -
Locked, Unlocked, Timeout (the time the door is allowed to be open)

KEYPAD -
Inactive, Door bell, Code entry, Code accepted

SOUNDER -
Off, Attention (door bell), Countdown (timeout), Alarm

DISPLAY -
The display only responds to the states of other elements and does not need to create any of its own states. Certain changes of state cause the display to be 'refreshed'

DISARM BUTTON -
The disarm button can be made available on the secure side of the door and is monitored by the keypad macro. When pressed, it momentarily forces the keypad to the same state as if a correct code had been entered. This allows the door to be opened from the secure side without causing an alarm condition (usual time limits apply).



Example program

The example program attached to this article supports all the functions described above.
The default code to gain entry is the coloured key sequence: blue, yellow, green, red. This can be easily changed in the first calculation block of the program.
When the door lock has been released, the door must be opened within 3 seconds. After opening, the door must be closed within 10 seconds.
Audible tones confirm the state of the system.
If the door is detected as being open at any time outside an entry sequence, the alarm will sound.

Expansion

This project can be combined with the code from the previous RC5 article to add remote access control functionality to the system. A simple remote lock release function could be implemented in a similar way to the disarm button (forcing a pseudo correct code condition in the keypad state)

The sounder used in the project is a fixed frequency piezo device. Alternative sounders might be able to use the PWM features of the MIAC to add tone variation to the audible indications.

The sounder can be replaced or augmented by indicator lamps.

Additional security devices (PIRs, pressure mats, etc.) can be added to the system to improve monitoring, or increase the number of zones being covered.

Multiple MIACs, and other devices, can be networked using the CAN bus port.

Further reading

Below are some links to other resources and articles on related subjects, and technical documentation relating to the hardware used for this project...

Flowcode:	http://www.matrixmultimedia.com/flowcode.php
MIAC:	http://www.matrixmultimedia.com/miac.php
Learning Centre:	http://www.matrixmultimedia.com/lc_index.php
User Forums:	http://www.matrixmultimedia.com/mmforums
Product Support:	http://www.matrixmultimedia.com/sup_menu.php