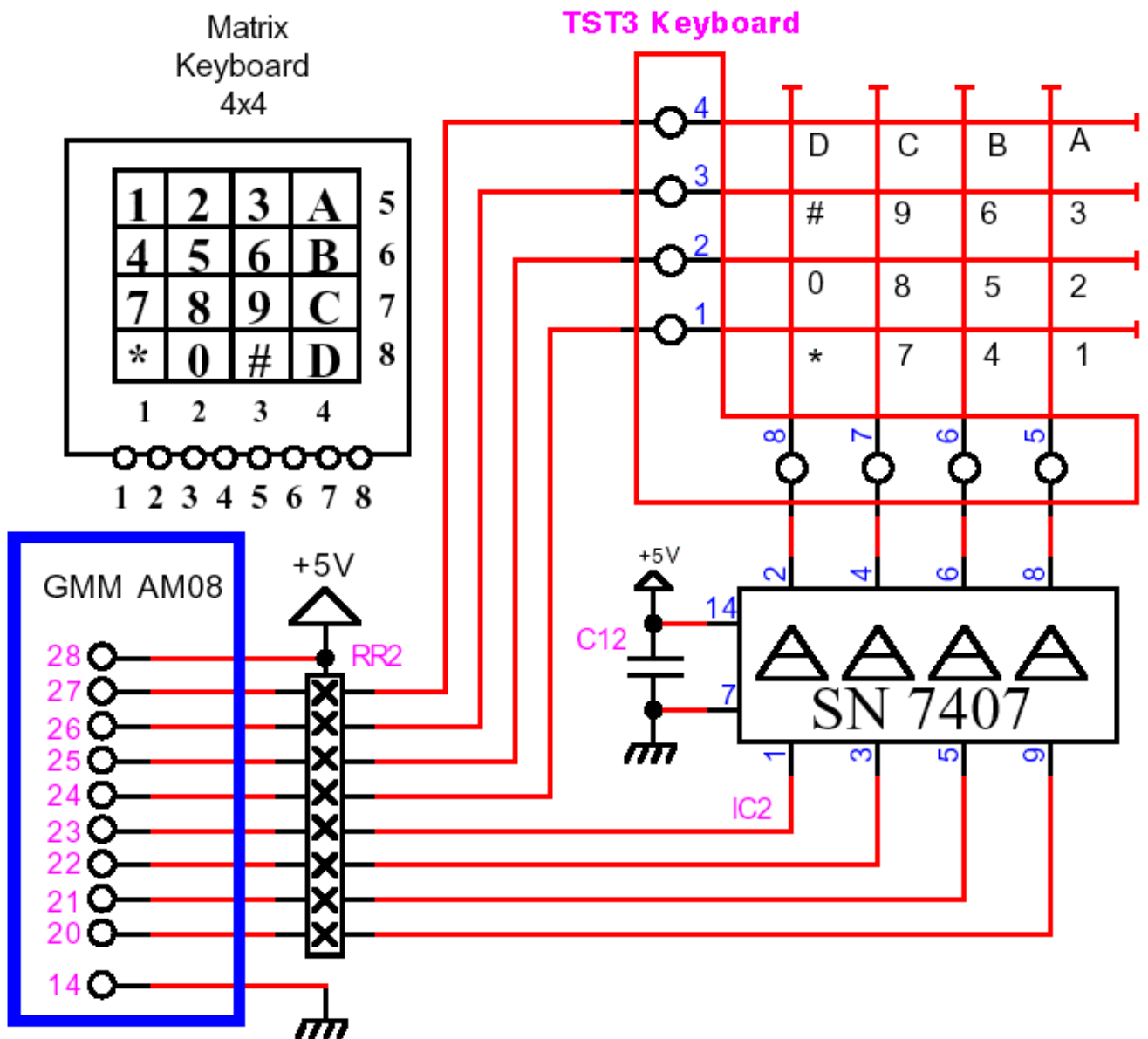


**KEYBOARDS (3).**

This program describes with details a complete solution of all the problems concerning the management of a **Matrix Keyboard**.



*Keyboard 4x4 Electric Diagram.*

The examined concepts show either how to acquire the keyboard with required securities of **Debouncing** (described in previous chapters), and the **Autorepeat** operations, when a key is continuously pressed.

By changing the timings defined in the program it is possible to change the operative modalities and so the user can adapt the keyboard functionality to different applications requirements.

The **Flow chart** at the end of chapter has been realized in order to explain the program parts and their relationship. It is composed by a general structure and some explosions of the most important components parts.

In addition to flow chart it is important to read the numerous remarks available in the program source. In fact, they show in details the used variables and constants and explain some strattagemmi used for autorepeat implementation.

## Example.019. Keyboard management. Debouncing and Autorepeat modality.

### Added Definitions:

None

### Added Declarations:

None

### Added Instructions:

None

### Added Operators:

None

**Example Program.019** of **BASCOM AVR** course.

It manages **all** keys of a **4x4** matrix keyboard.

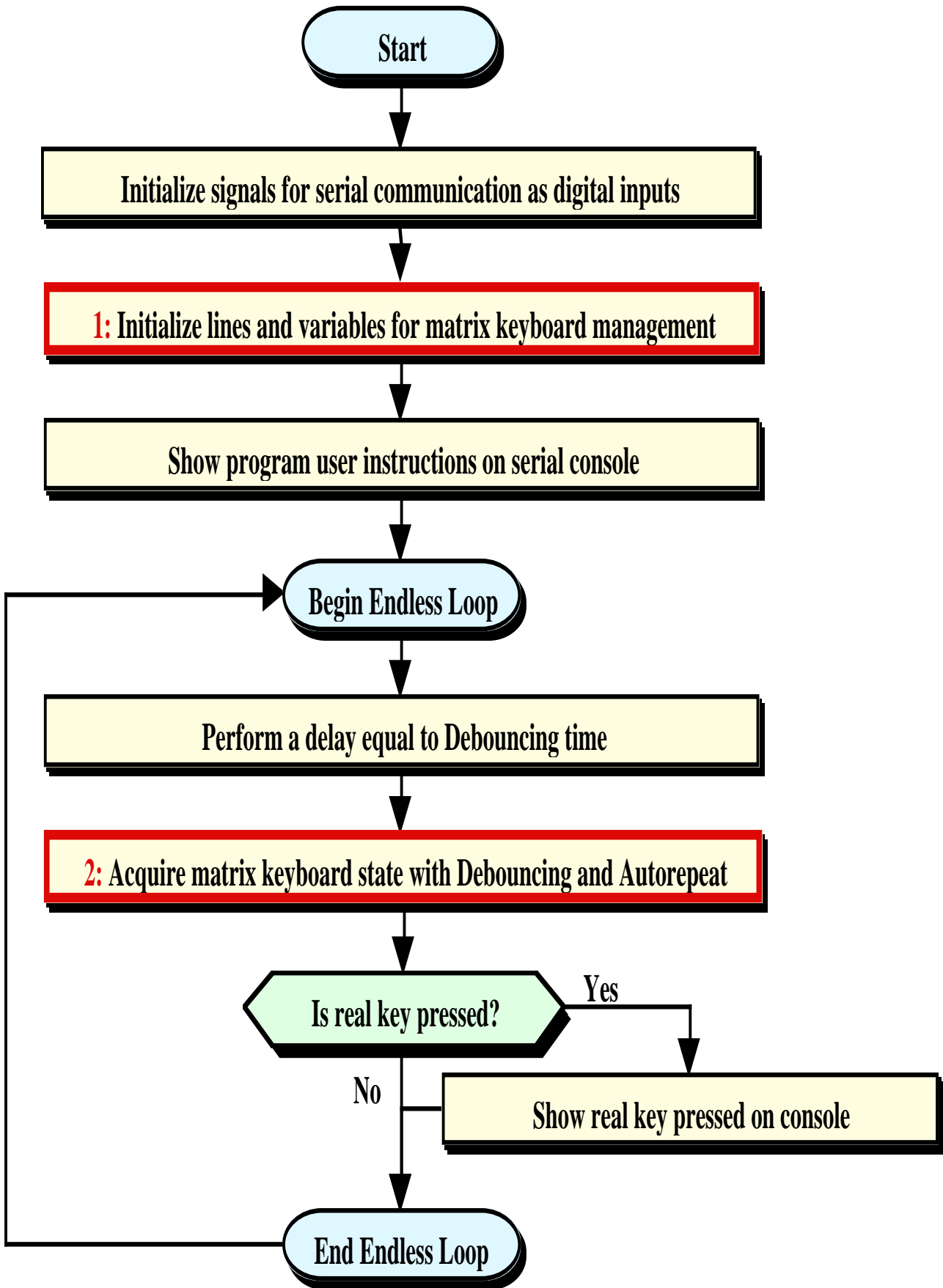
The program continuously acquires the state of the **16** keys connected to matrix keyboard available on **GMM TST3** and it transmits them on serial line. The acquisition is performed with **Debouncing** and **Autorepeat**: the keys are acquired with a **Debouncing** time of **20** milliseconds, the first key with **Autorepeat** is supplied after **500** milliseconds and the following ones after **100** milliseconds.

The visualization of pressed keys is performed on a serial console provided of monitor and it must communicate with a fixed physical protocol at **19.200 Baud, 8 Bit x chr, 1 Stop bit, No parity**.

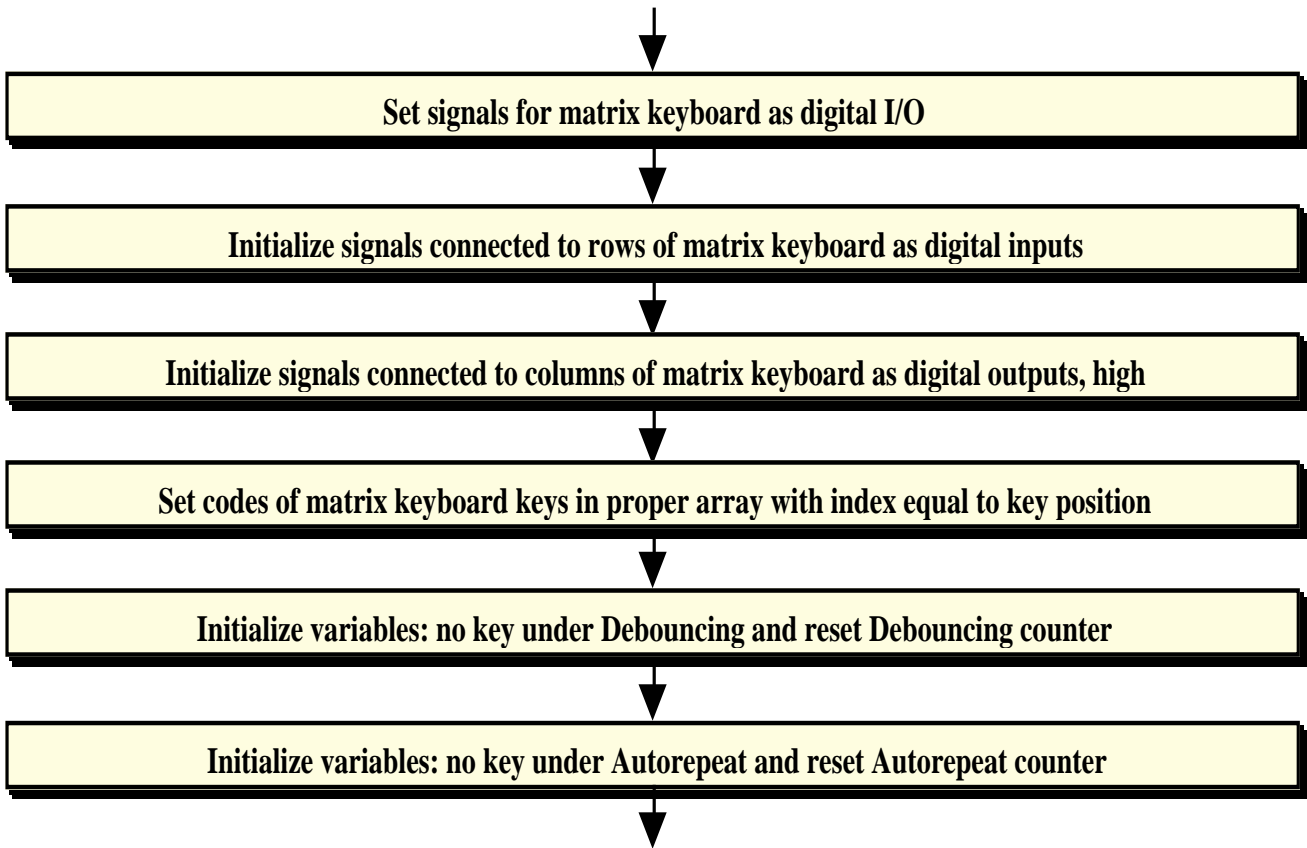
This console can be another system capable to support a serial **RS 232** communication. In order to simplify the use it can be used a **PC** provided of one **COMx** line, that execute a terminal emulation program as **HYPERTERMINAL** or the homonym modality provided by **BASCOM AVR** (see **IDE Configuration**).

The program works only when the **GMM AM08** is mounted on **Z1** socket of **GMM TST3!!**

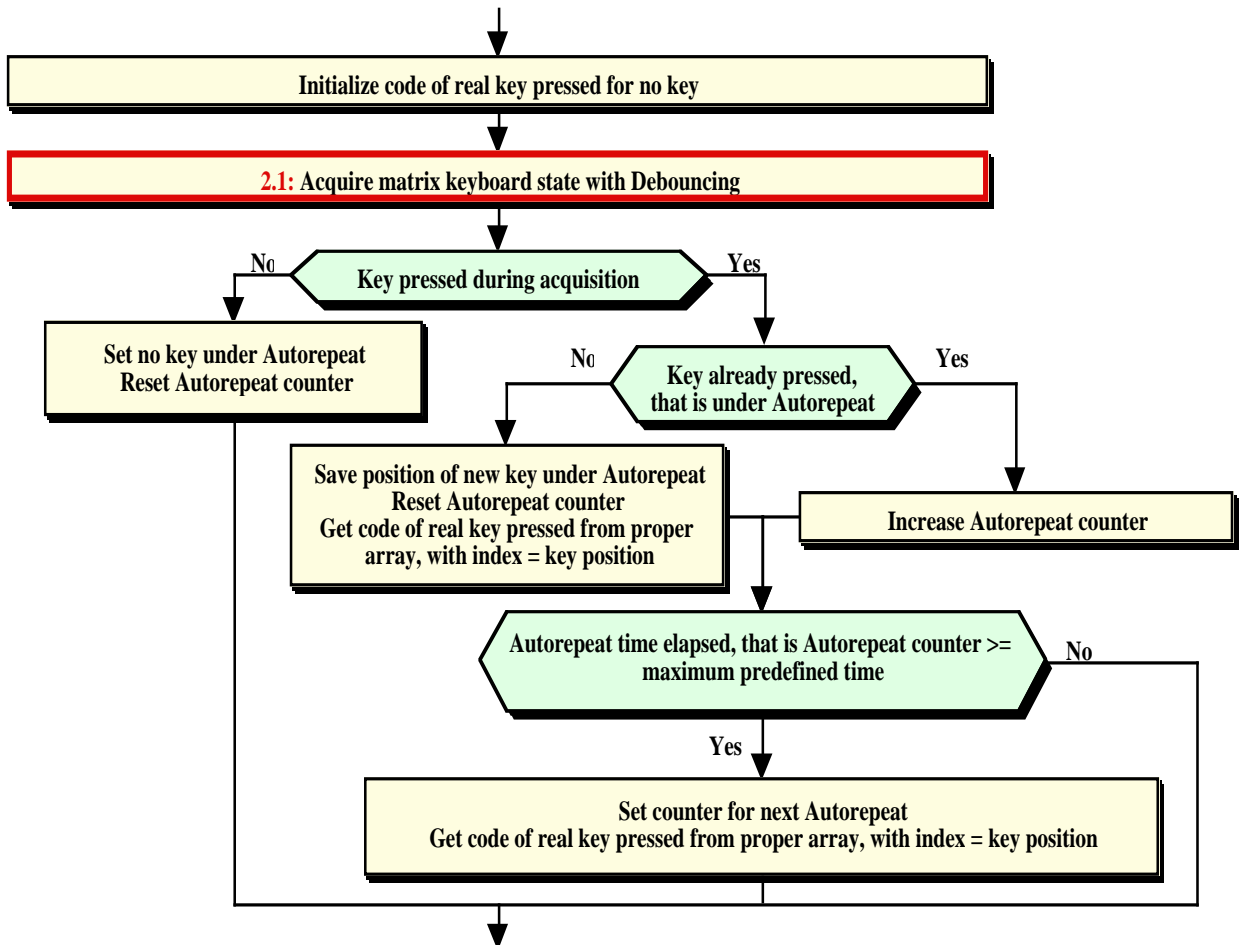
Inside the program the **Row** and **Column** terms refer to electric diagram of matrix keyboard, not to its physical format!!



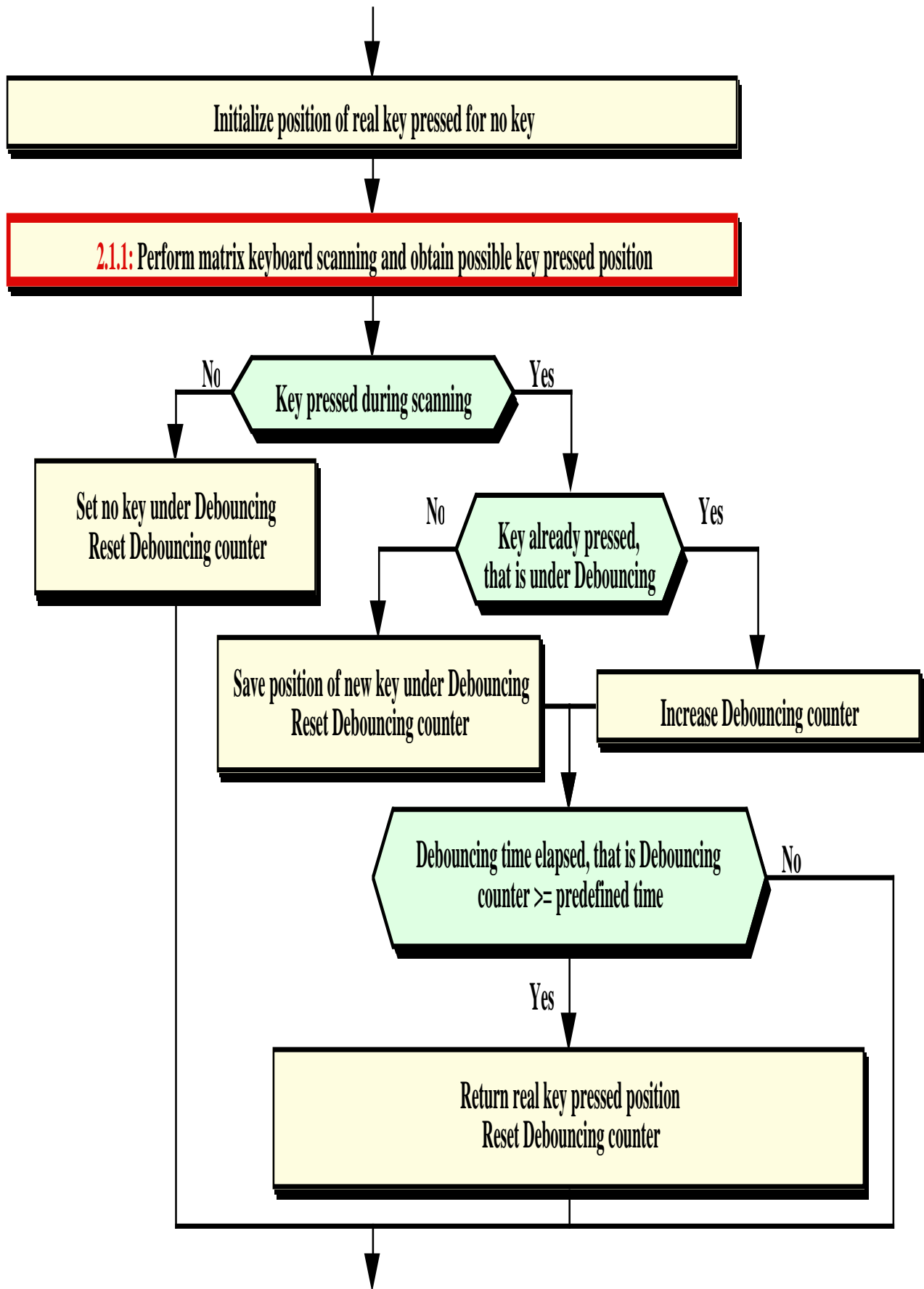
*Flow Chart Diagram of the Program.*



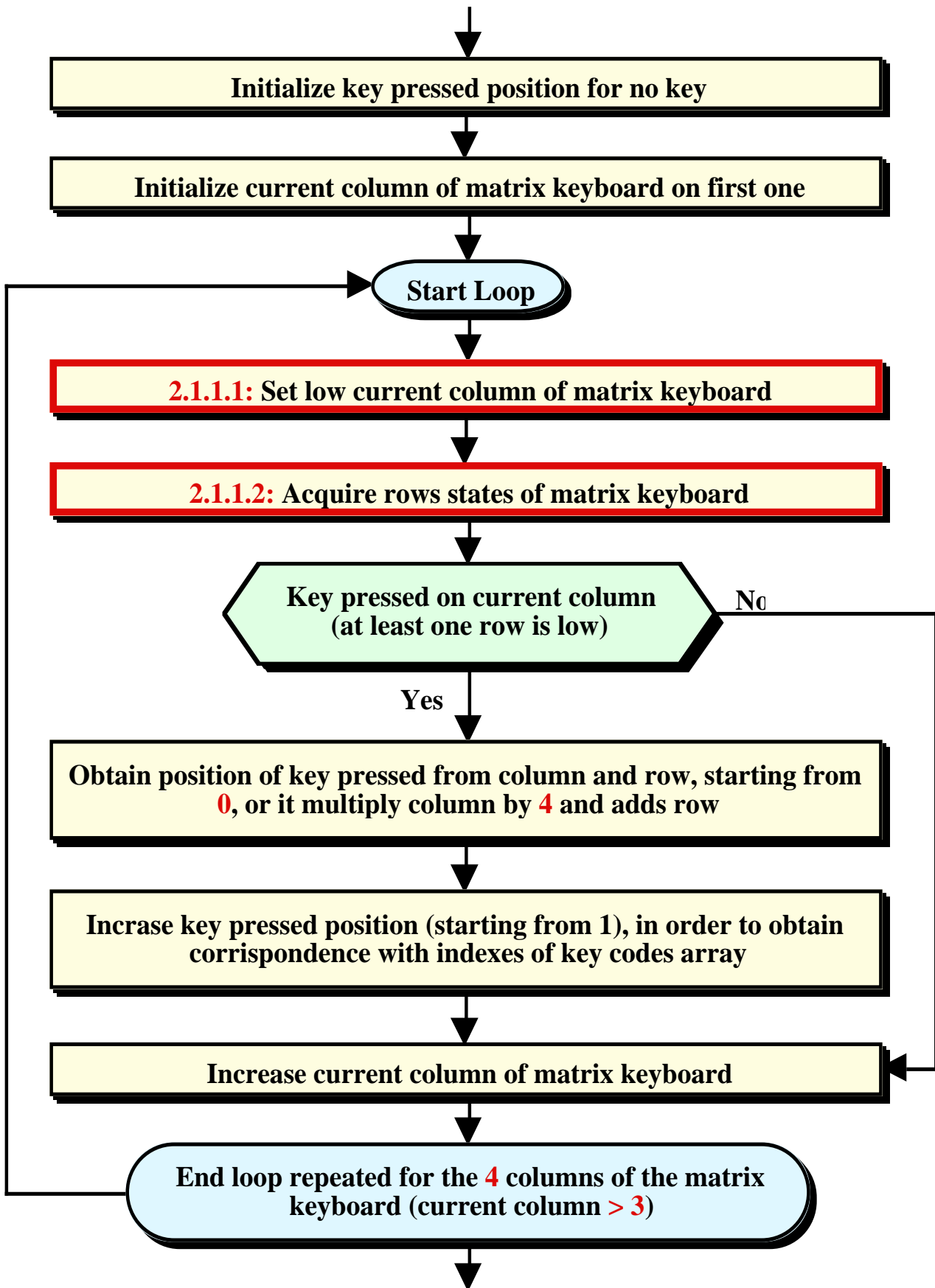
**1: Initialize Lines and Variables for Matrix Keyboard Management.**



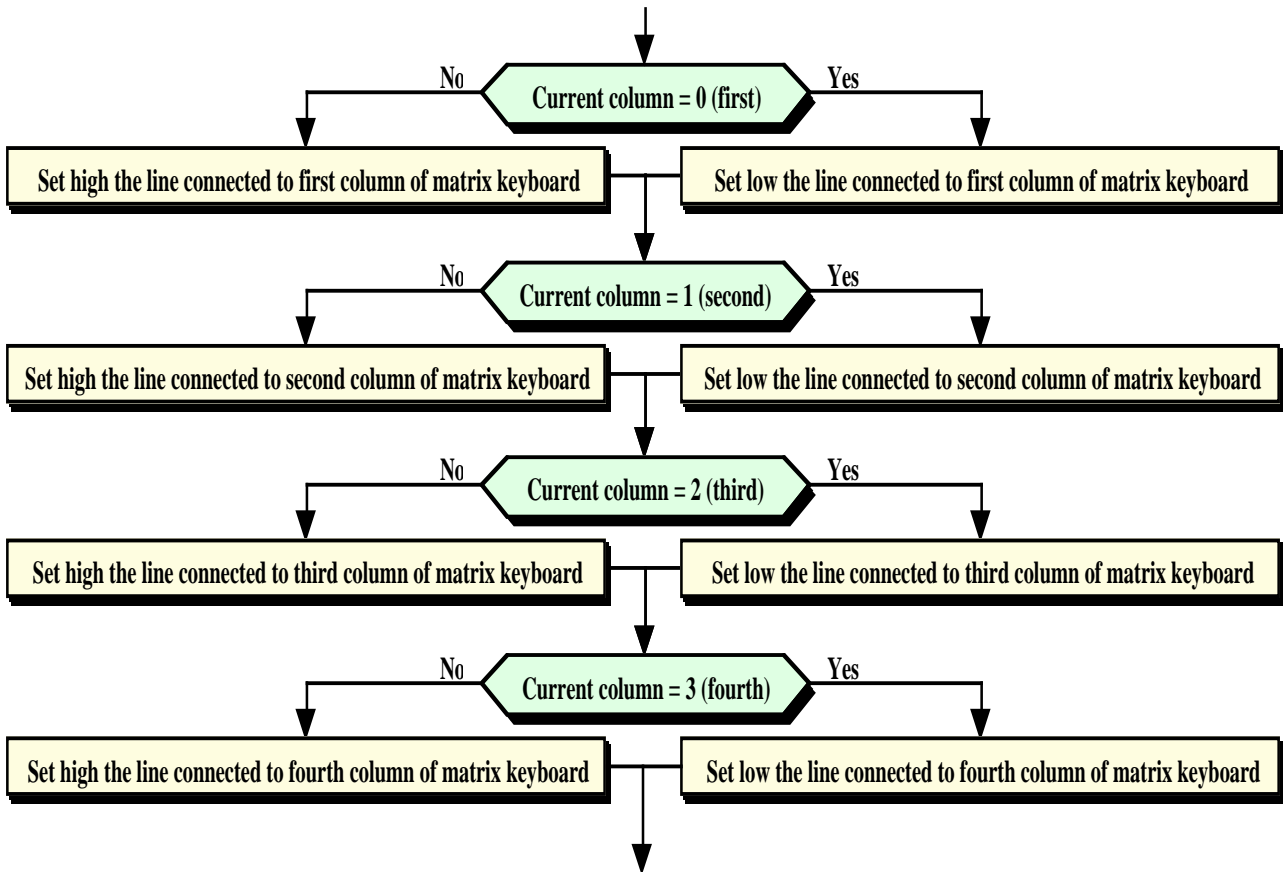
**2: Acquire Matrix keyboard State with Debouncing and Autorepeat.**



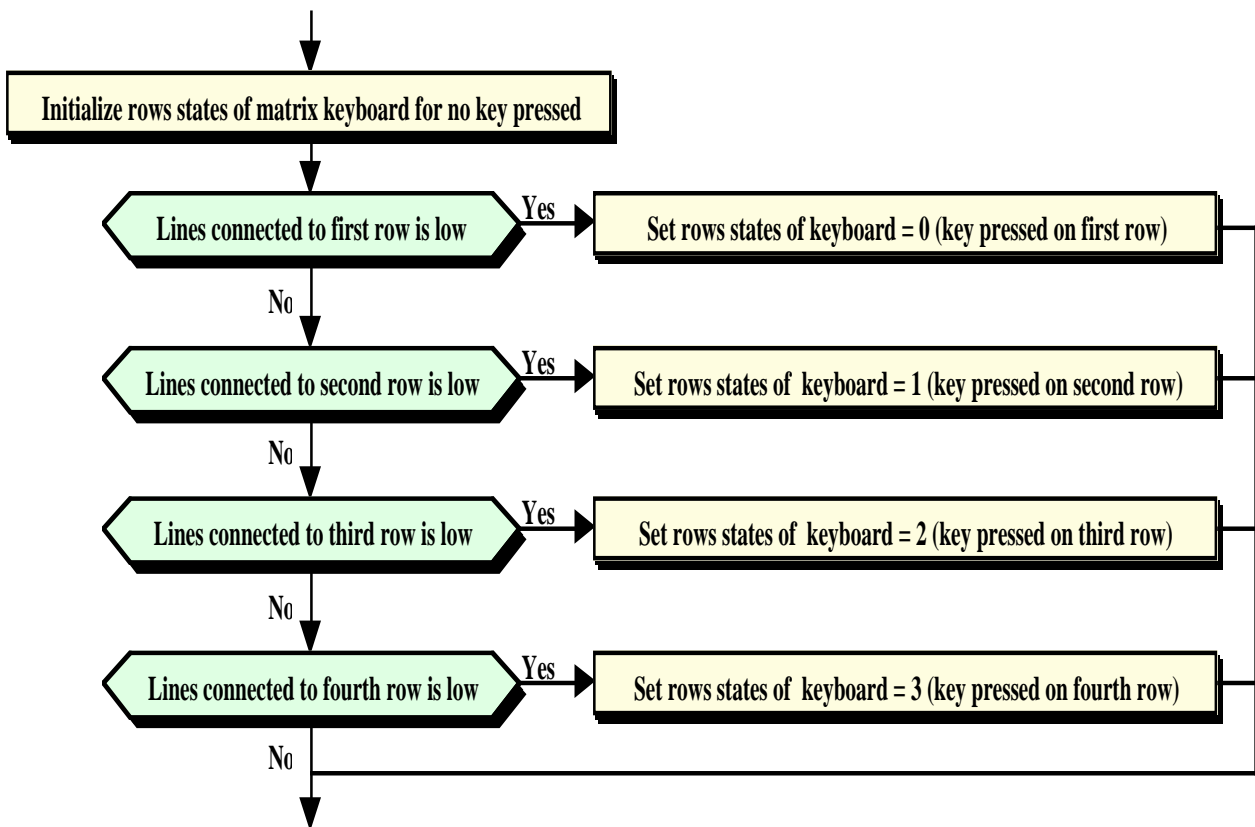
**2.1: Acquire matrix keyboard state with Debouncing.**



**2.1.1: Perform Matrix Keyboard Scanning and obtain Possible Key Pressed Position.**



**2.1.1.1: Set Low Current Column of Matrix Keyboard.**



**2.1.1.2: Acquire Rows State of Matrix Keyboard.**