

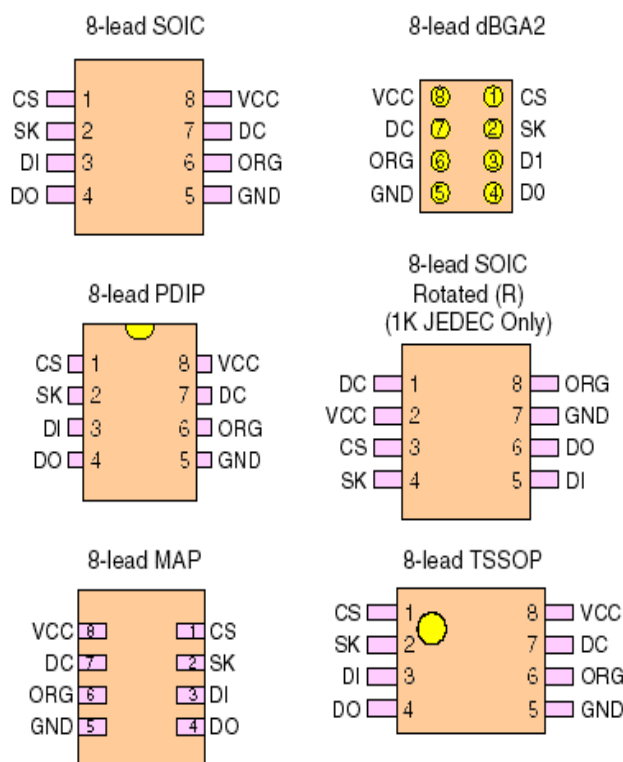
*Theoretic/Practical course on BASCOM AVR Programming.
Author: DAMINO Salvatore.*

MICROWIRE® SERIAL EEPROM.

In **Embedded** applications, among the most frequently used devices, there are the **Serial EEPROM**. These devices are available with different memory sizes, different packages and moreover with some, and different, **Communication Protocols**. In this chapter we'll describe the **Microwire® Protocol**.

Table 1. Pin Configurations

Pin Name	Function
CS	Chip Select
SK	Serial Data Clock
DI	Serial Data Input
DO	Serial Data Output
GND	Ground
VCC	Power Supply
ORG	Internal Organization
DC	Don't Connect

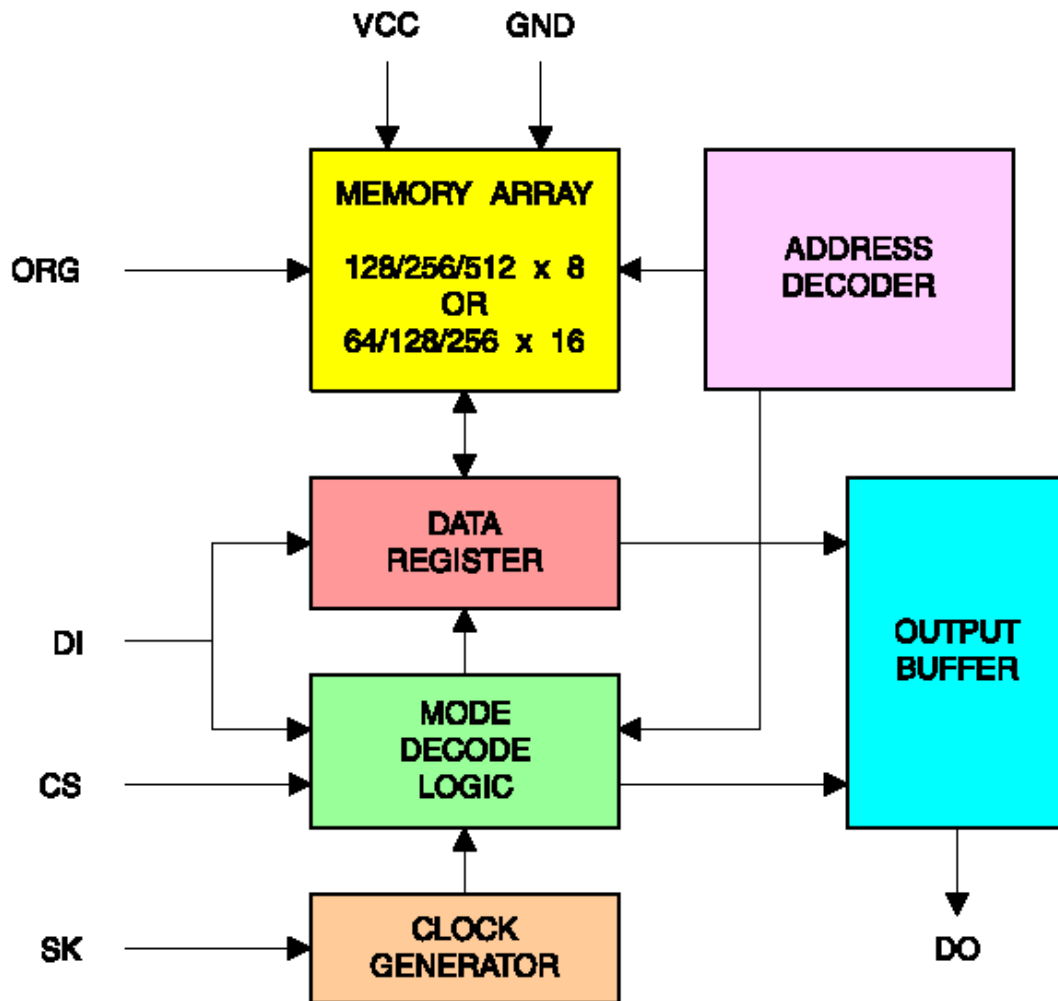


Microwire® Serial EEPROM 93LC46A.

The **EEPROM** are **memory, not Volatile**: this means that they maintains the stored information even without power supply. The devices can be **written** a very high number of times and they can be **read** without limits, as standard **RAM**.

The management operations are defined by accurate rules, that are described below.

The internal structure is reported in the following **diagram** that lists all the logic blocks included in the device.



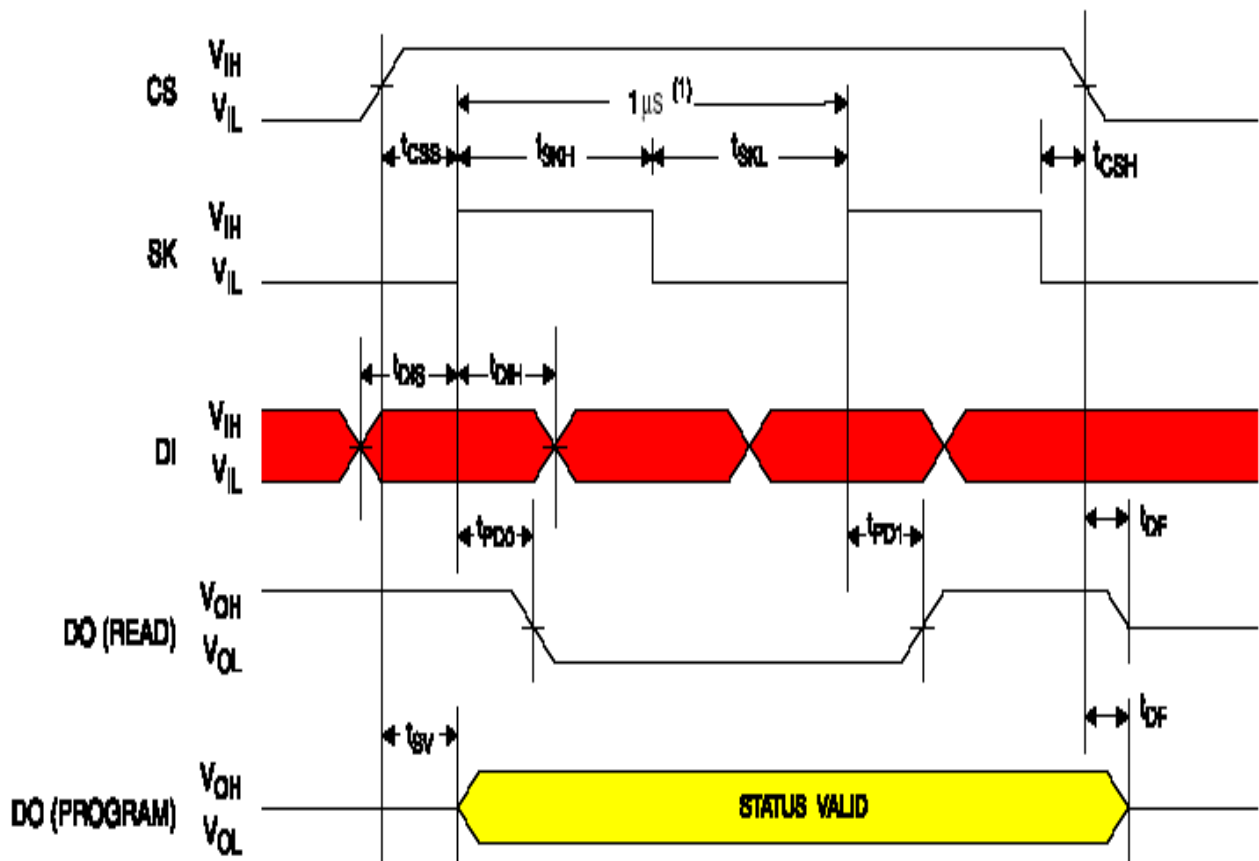
Blocks Diagram of Microwire® Serial EEPROM.

The communication is performed through **4** signals: the first (**CS**) enables the **EEPROM**, the second supplies **timings** and **synchronizations** (**SCK**), the third sends data to device (**DI**) joined with the fourth that contemporaneously acquires data from the device (**DO**).

Between the data sent to **EEPROM**, over the address (that specifies the location to use) and the data to write, there are also some commands that select the operation to perform.

In order to get a detailed and bright vision of the device, we suggest You firstly to read and study the specific component **Data Sheet** and then to try the developed examples.

The **Microwire® EEPROM** are widely used in automotive sector and in the **white electronics** field (washing machines, refrigerators, etc.) and they main scopes are the Kilometers counting, mantain the number of copies performed by a Fax or a Photocopier, store the total worked hours, etc.



Note: 1. This is the minimum SK period.

Synchronous Data Timing

Timing Diagram.

Example.047. Byte R/W management of a serial EEPROM 93LC46A

Added Definitions:

None

Added Declarations:

None

Added Instructions:

None

Added Operators:

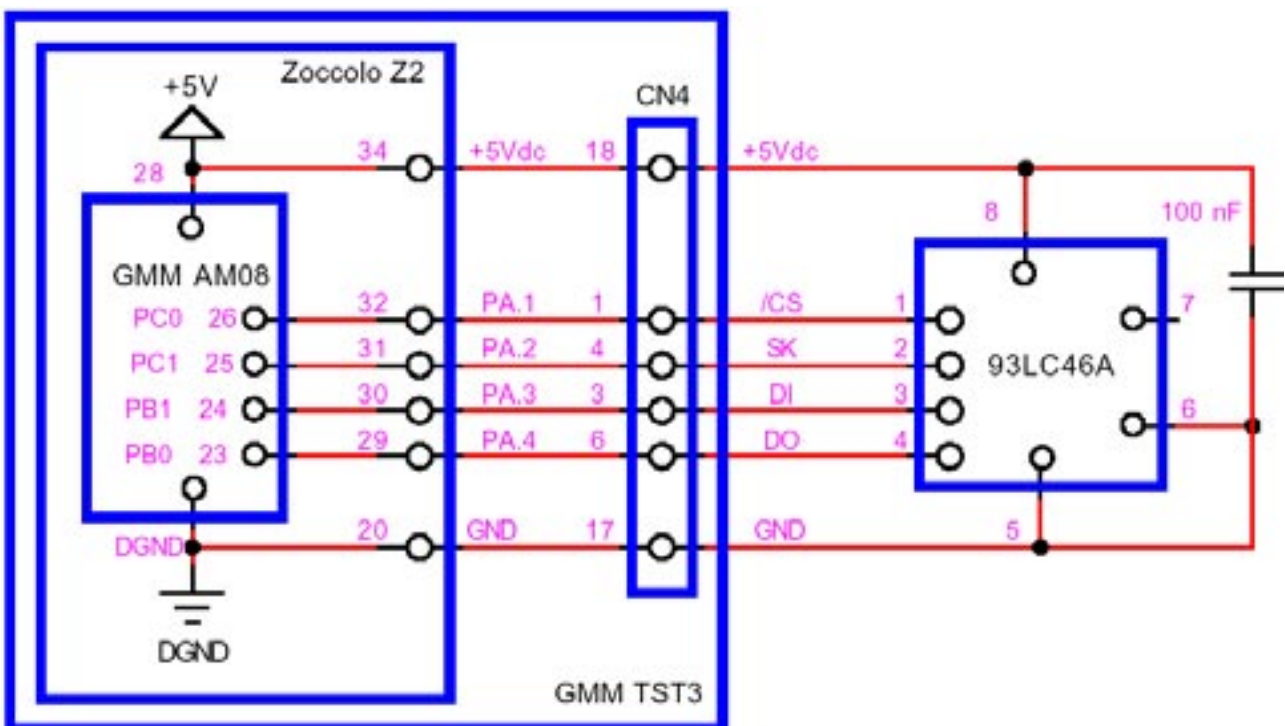
None

Example program 47 of **BASCOM AVR** course.

Test and management program for **Microwire EEPROM 93LC46A**.

A peculiar feature of this device is the requirement of a write enable operation before any attempt to store data.

This feature acts as a protection against accidental write, or erase operations.

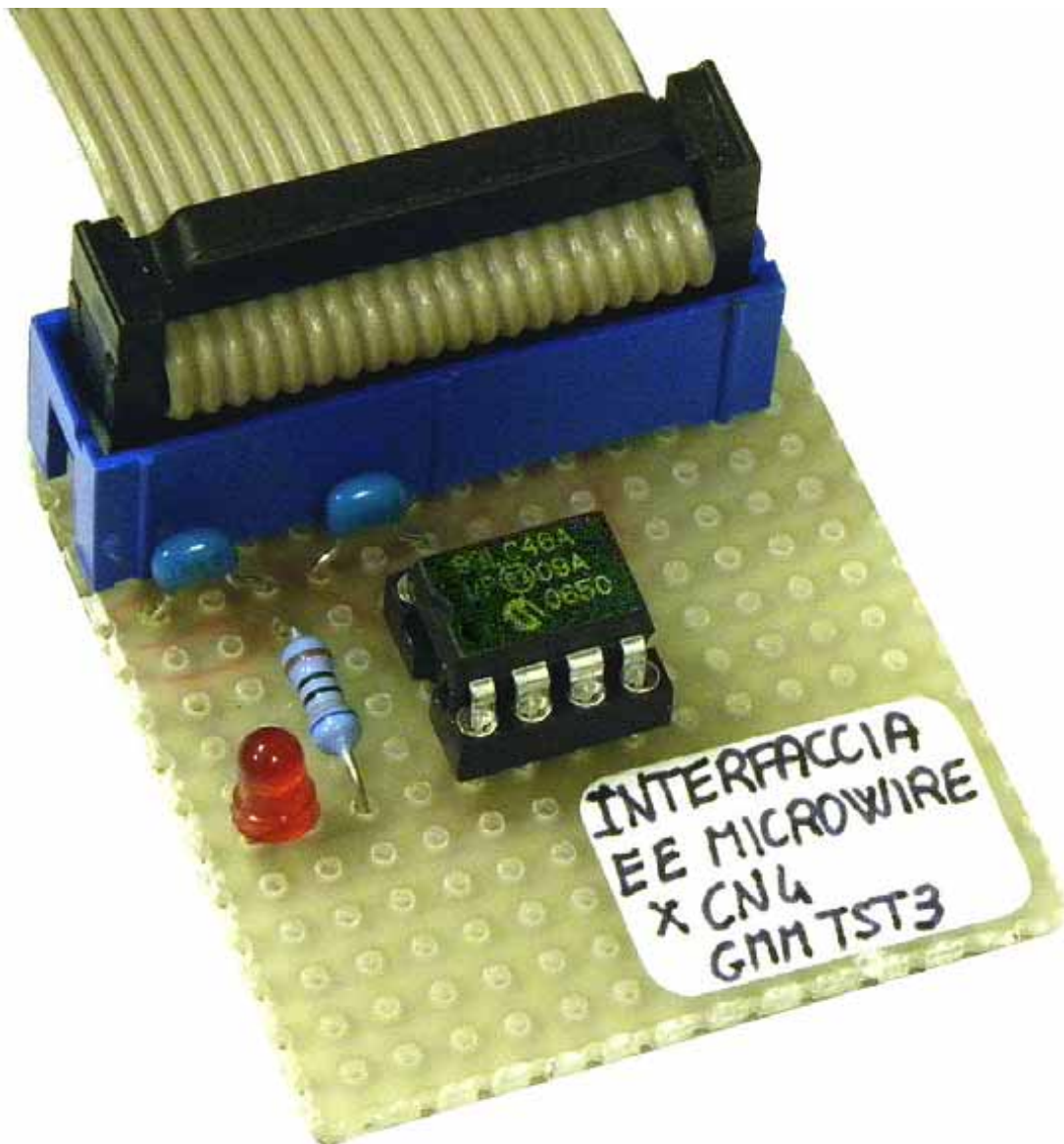


Electric Application Diagram of a Microwire® Serial EEPROM 93LC46A.

It performs the fundamental operations on the component by using a software **Microwire** interface and by interacting with user on a serial console provided of monitor and keyboard with a fixed physical protocol at **192.00 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 **Z2** socket of **GMM TST3!!**



Experimental Card, on Prototype Board, of Microwire® Serial EEPROM.

Example.048. R/W management of a serial EEPROM 93LC46A with messages long up to 20 characters, displayed on Console

Added Definitions:

None

Added Declarations:

None

Added Instructions:

None

Added Operators:

None

Example program 48 of **BASCOM AVR** course.

Messages read and write on **Microwire EEPROM 93LC46A**.

It performs the operations on the component by using a software **Microwire** interface and by interacting with user on a serial console provided of monitor and keyboard 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 **Z2** socket of **GMM TST3!!**