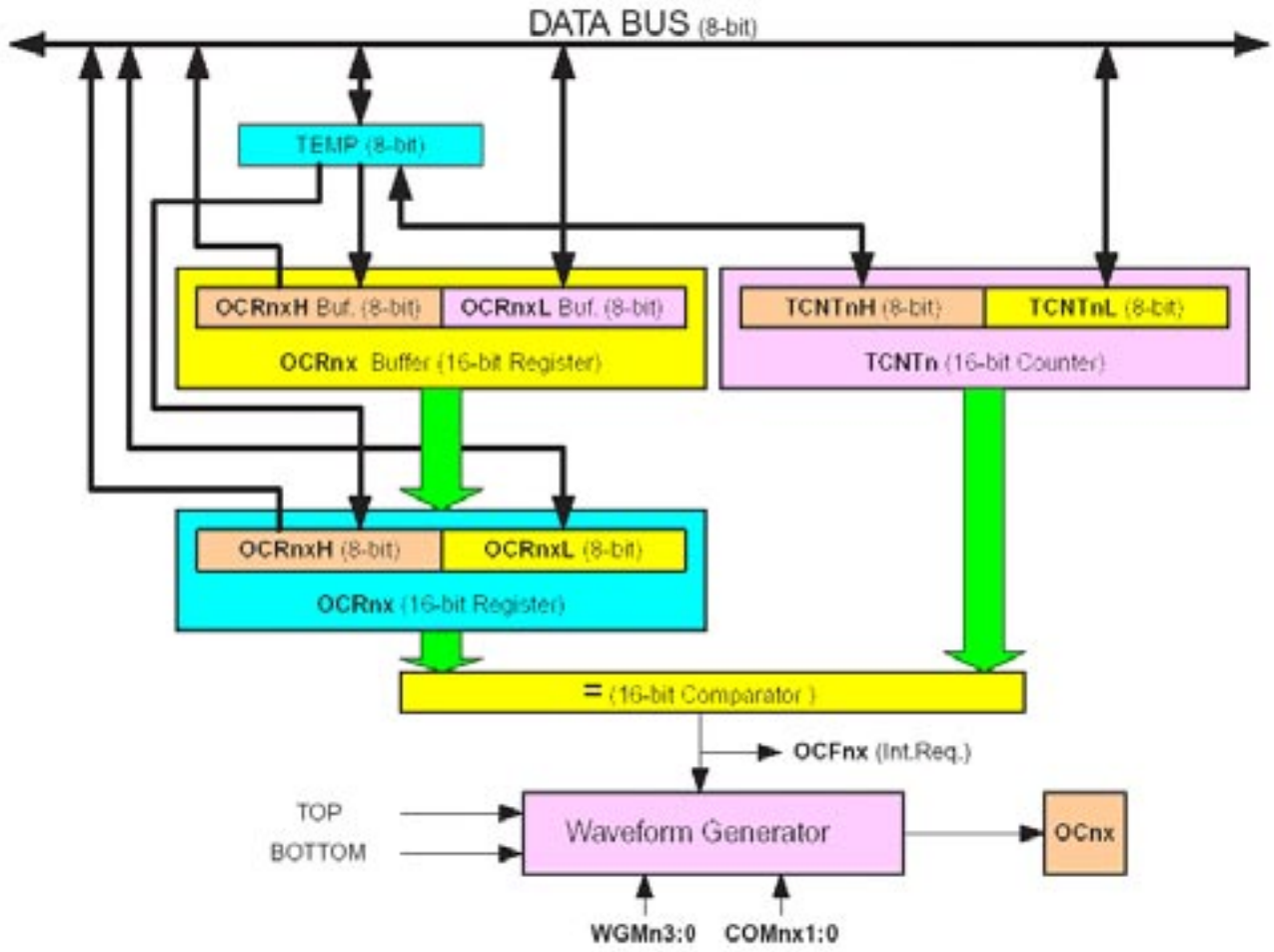
PWM Section and D/A Converter Generator.

The internal structure of **Mini Modules** is composed by many different sections capable to offer so many **Hardware** interfaces that support the majority of possible applications. Among the available units inside **Mini Modules**, a really interesting section is those named **PWM**.

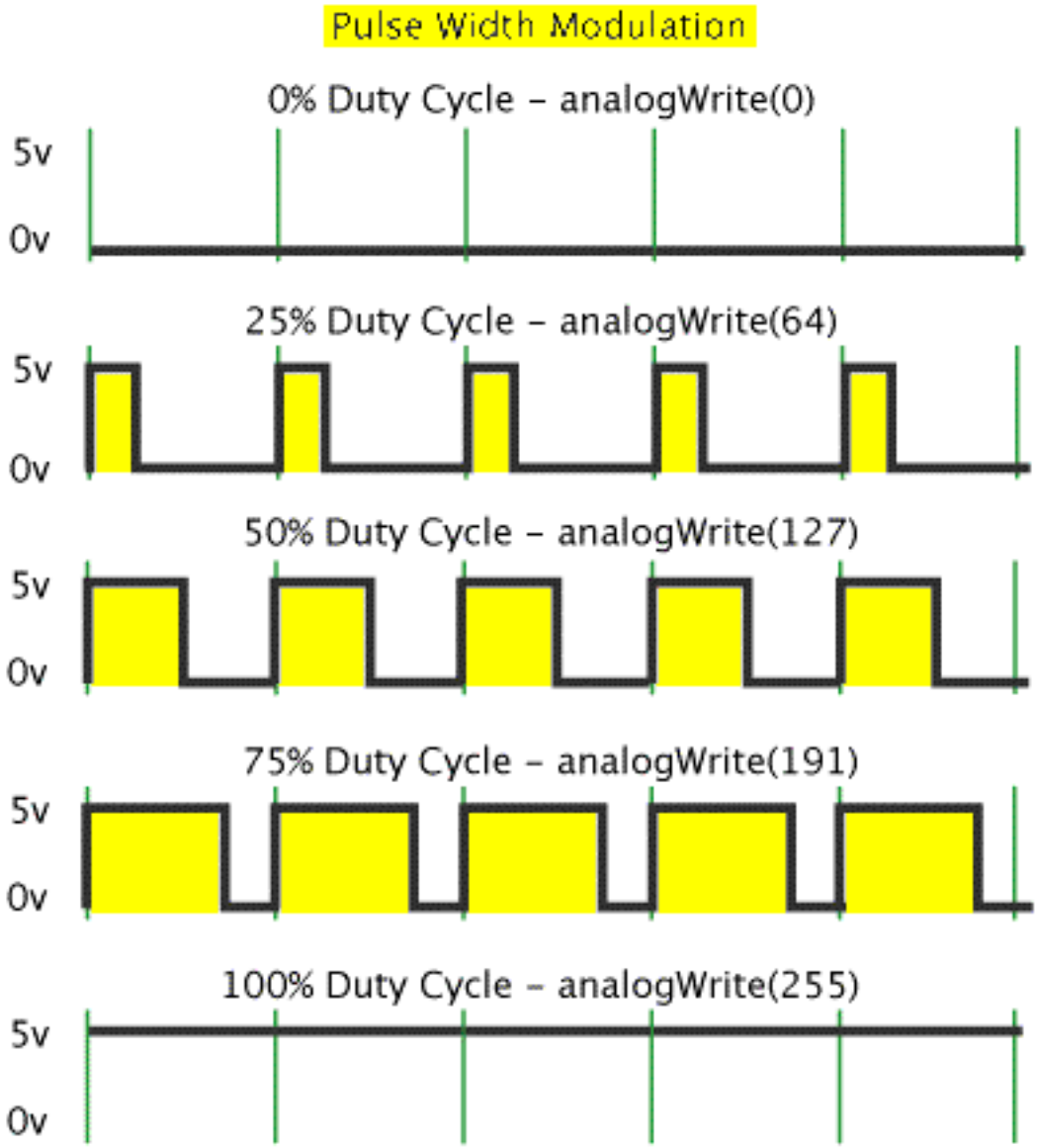


Block Diagram of PWM Section.

The **Pulse-Width Modulation** or **PWM**, is a type of **Analog Modulation** where the information is coded with a time duration of each pulses of the signal.

The duration of each pulse can be defined as the ratio between the **2** periods of the consecutive levels of the pulse, that is the **Duty Cycle** notion.

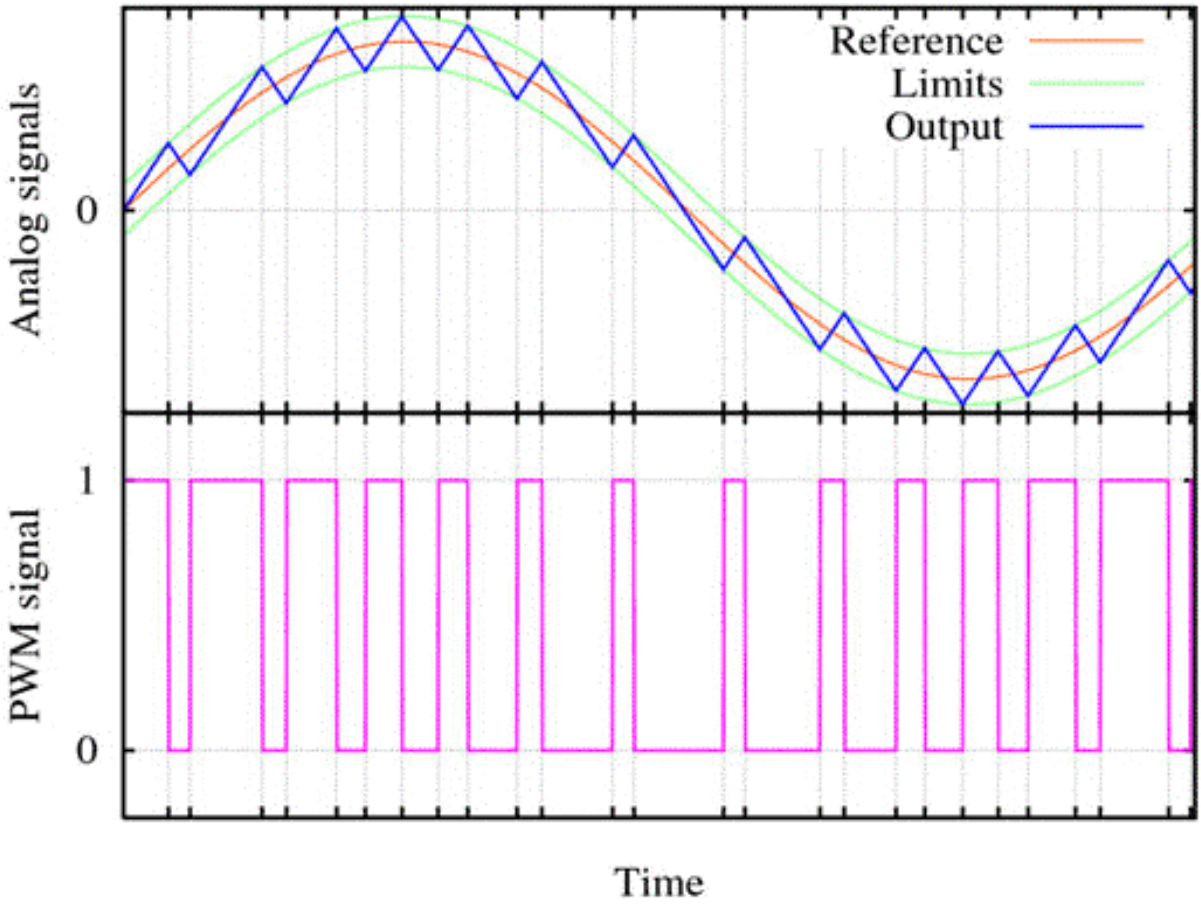
A **Duty Cicle** equal to **0%** denotes pulse with null duration, that in practice is a null signal, while a **100%** value denotes a pulse that stops exactly when the following one starts.



PWM Signal Waweforms.

Many times the designers must generate analog voltage levels in electronic circuits that are completely digitals.

Even if the market offers a really wide range of suited **Digital to Analog** converters, it is possible to adopt also a low cost solution, by using the **PWM** section of **Mini Module**.



Conversion of PWM Signal Into Analog Signal.

The previous figure shows how a **PWM** signal is converted into Analog signal. By using this technique, and only few passive components, you can generate a good **Digital/Analog** converter.

In the following examples you can examine some programs that use a **Mini Module** and they exploit the **PWM** output line in order to generate one analog voltage capable to cover the full range from **0 Vdc** to power supply voltage, that is the maximum generable value.

By using the supplied examples, and a standard **Volt meter** or **Tester**, it is possible to immediately visualize the generated voltage value.

This technique can be adopted in many applications. The most diffused is, for example, the connection to an input of a power actuator that drives a **DC motor**, or a proportional valve, etc.

This solution, really inexpensive, allow to drive small actuators that move mechanisms, where the resolution, reached by **PWM** signal, is sufficient.

Example.063. PWM Management. It Generates a D/A Conversion Output, Based on a PWM.

Added Definitions:
None

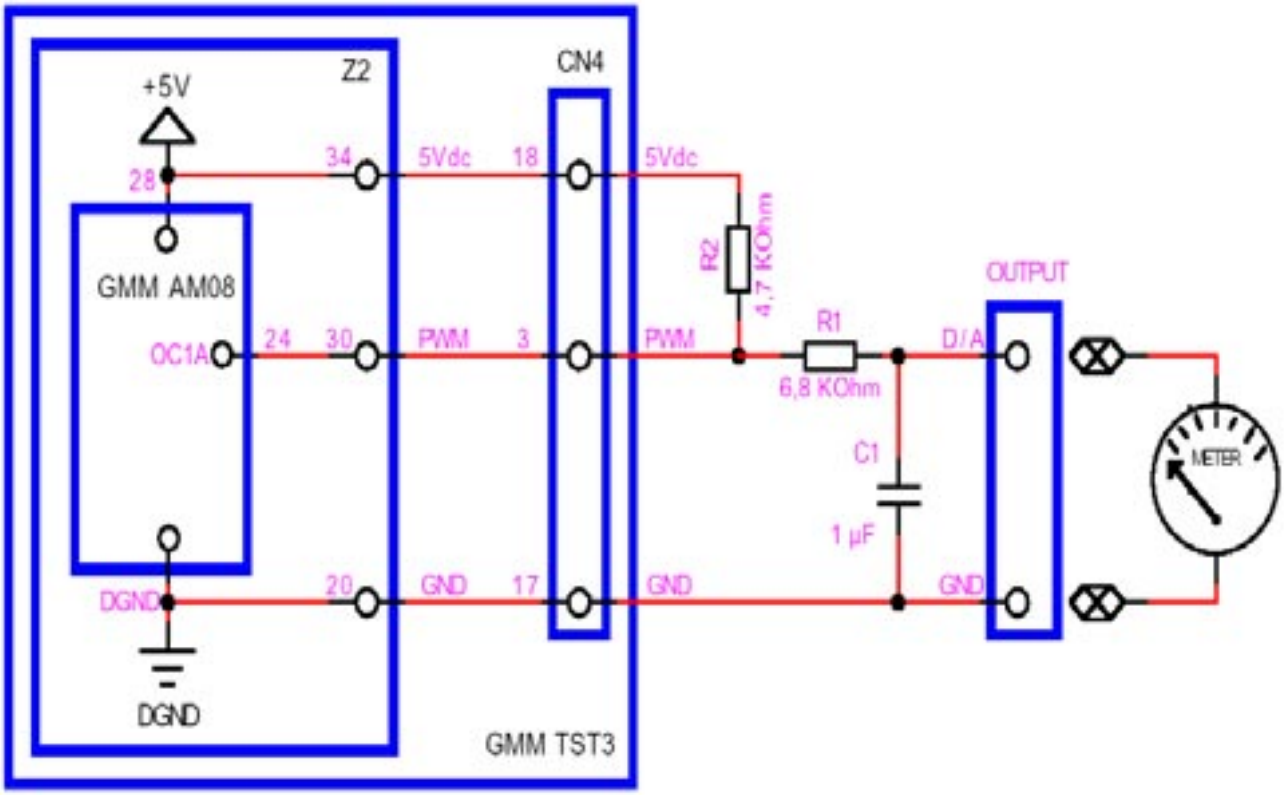
Added Declarations:
None

Added Instructions:
CONFIG TIMER1; PWM1A.

Added Operators:
None

Example program **063** of **BASCOM AVR** course.

PWM management: it generates a **D/A Conversion** output, based on a **PWM** signal.



Generate a D/A Converter Through a PWM Line.

The program generates a **PWM** signal with a fixed frequency of about **14.500 Hz** and a **Duty Cycle** settable by user, through console. The generated Pulse Width Modulation signal, when connected to proper **RC** circuit, it produces an analog signal variable in **0÷5 V** range.

The **PWM** signal is generated by hw from the **Timer 1** section of microcontroller, where the compare signal connected to pin **OC1A** of **Mini Module**, reported on connector **CN4.3** of **GMM TST3**, as described in the electric diagram.

The resolution used on **Timer 1** is **8** bits and the generated **PWM** signal will have **256** different values of duty cycle.

The program describes its functionalities and uses 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**!!

Example.064. PWM Management: It Generates a D/A Conversion Output, Continuosly Increased and Decreased, Based on a PWM Signal.

Added Definitions:
None

Added Declarations:
None

Added Instructions:
STEP.

Added Operators:
None

Example program **064** of **BASCOM AVR** course.

Gestione PWM: Genera un'uscita di conversione **D/A**, continuamente crescente e decrescente, tramite un segnale **PWM**.

PWM Management: it generates a **D/A** conversion output, continuosly increased and decreased, based on a **PWM** signal.

The program generates a **PWM** signal with a fixed frequency of about **14.500 Hz** and a **Duty Cycle** continuosly changed from the minimum to maximum and after from maximum to minimum. The generated **Pulse Width Modulation** signal, when connected to proper **RC** circuit, it produces a triangle waveform analog signal variable in **0÷5 V range**. The selection of the **RC** circuit components defines both the stability of generated **D/A** signal and the response time in **Duty Cycle** variation following. So, these values change according with user requirements and load connected to **D/A** signal.

The **PWM** signal is generated by **hw** from the **Timer 1** section of microcontroller, where the compare signal connected to pin **OC1A** of **Mini Module**, reported on connector **CN4.3** of **GMM TST3**, as described in the electric diagram.

The resolution used on **Timer 1** is **8** bits and the generated **PWM** signal will have **256** different values of **Duty Cycle**.
The program describes its functionalities and uses 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**!!

Example.065. PWM Management: It Drives a LED Brightness Through a D/A Conversion Line. The D/A Line is Driven by the Position of a Potentiometer, Acquired Through A/D Converter.

Added Definitions:
None

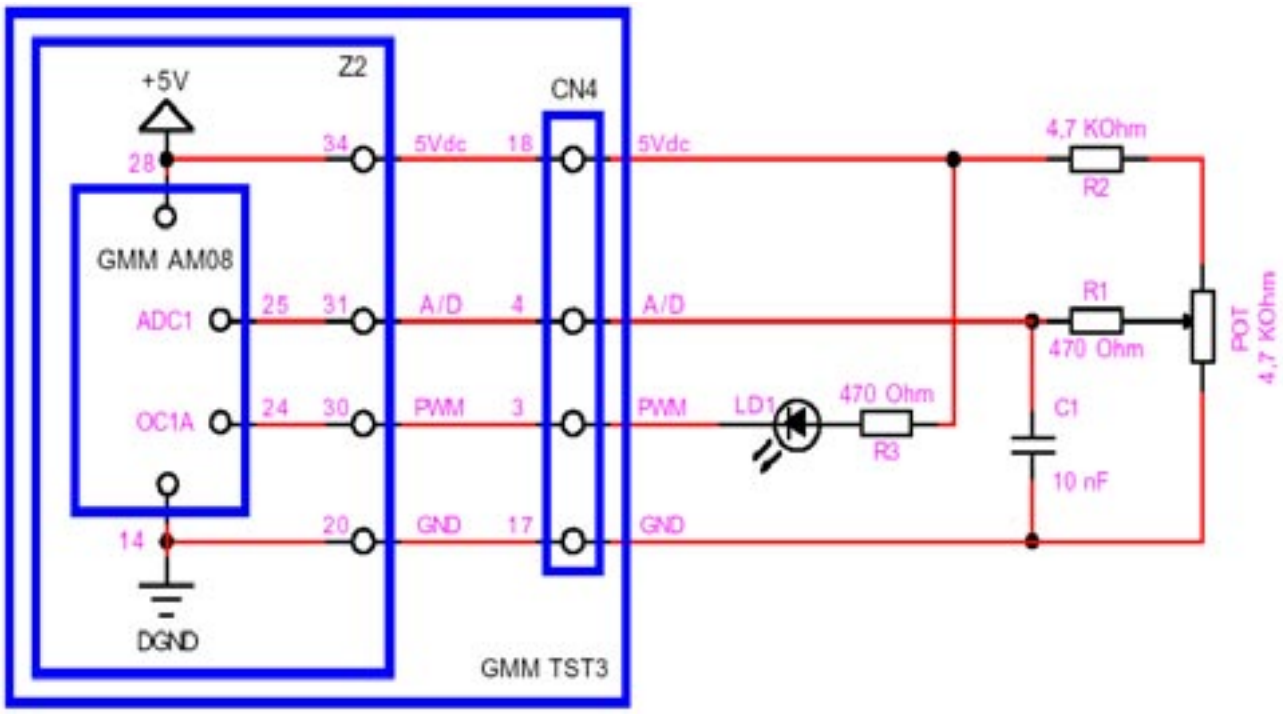
Added Declarations:
None

Added Instructions:
None.

Added Operators:
None

Example program **065** of **BASCOM AVR** course.

PWM management: it generates a **D/A** conversion output, based on a **PWM** signal, defined by the position of a **potentiometer**, acquired through **A/D** converter.



D/A Converter Driver by a Potentiometer.

The program generates a **PWM** signal with a fixed frequency of about **14.500 Hz** and a **Duty Cycle** changed from the **potentiometer** position.

The generated **Pulse Width Modulation** signal, when connected to proper **RC** circuit, it produces an analog signal proportional to potentiometer position, variable in **0÷5 V range**.
The selection of the **RC** circuit components defines both the stability of generated **D/A** signal and the response time in **Duty Cycle** variation following. So, these values change according with user requirements and load connected to **D/A** signal.

By connecting a **LED** to **D/A** signal, its brightness will change according to potentiometer variations and, as every thing is driven by software, the program can decide the modality. For example the **LED** brightness could be defined even from digital inputs, delays, other analog inputs, temperatures, parameters in memory, etc.

The **PWM** signal is generated by **hw** from the **Timer 1** section of microcontroller, where the compare signal connected to pin **OC1A** of **Mini Module**, reported on connector **CN4.3** of **GMM TST3**, as described in the electric diagram.

The resolution used on **Timer 1** is **8 bits** while the **A/D** section resolution is **10 bits**: the program will adapt the different values.

The program describes its functionalities and uses a serial console provided of monitor 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**!!