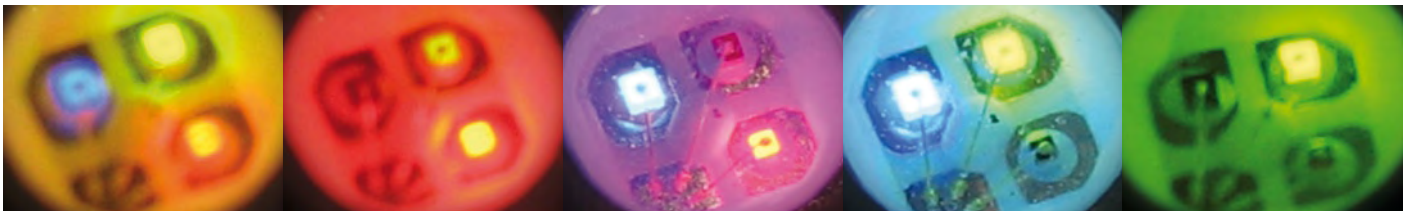
RGB LED Driver

High resolution colour control with the AAT3129



Fred Splittgerber (Germany)

Seemingly straightforward projects can turn into a 'money pit' or 'component graveyard' if you are not careful. This can easily come true if you intend driving colour LEDs in RGB mode with infinitely variable colour mixing and individual control over the brightness of each LED. Conventional control circuitry tends to produce quite bulky systems too. On the other hand, using a microcontroller and a specialised IC keeps the space footprint under control and eliminates all the uncertainties...

Specifications

- RGB LED driver module for universal application
- Straightforward serial control
- Operating voltage from 2.7 V upwards thanks to charge pumps
- LED maximum current (total) 40 to 180 mA
- 4,096 colours
- 16 stages of total brightness
- Low-interference operation at constant frequency
- Flicker-free illumination thanks to 1 MHz PWM frequency

Listing every possible application for infinitely variable control of individual RGB LEDs is an impossible task. What is not in dispute is the fact that the variety of RGB LEDs (one each in red, green and blue on a single carrier or in a single package) has risen significantly in recent years. Anyone planning to put these colourful semiconductor light sources to practical use needs to think carefully about the control electronics to be used.

RGB control

The rules covering LEDs in general apply also to RGB LEDs, the most fundamental being that LEDs need powering with constant current rather than constant voltage. This is because the threshold voltages of LEDs are strictly temperature-dependent and without constant current, stable operation is impossible. Simple logic indicates that achieving infinitely variable (step-free) current setting requires the use of infinitely variable current sources. If energy saving is important, then the recommended approach is to use switched constant current sources with adjustable duty cycles.
An important characteristic of RGB LEDs to note is that as a result of their physical structure, red, green and blue LEDs display differing forward voltages, ranging from less than 1.5 V for red LEDs up to nearly 4 V for blue ones. Without some kind of intelligent switching arrangement it's obvious that significant energy losses will arise if your driver circuitry provides the same voltage for R, G and B LEDs (which will be far too high for the red ones). Pulse-width modulated current sources are totally unsuitable, especially in battery powered applications. But before you bash your brains in looking for suitable solutions based on switching regulators, take it easy. Industry has already come up with a solution for this problem and embedded it in silicon.

AAT3129

As well as its switching regulators and other power supply ICs, the firm

Analogic Tech has lately brought out a whole range of chips intended to simplify the operation of all manner of LEDs. And with the IC AAT3129 [1] every control problem that might occur with RGB LEDs has been eliminated with a single chip.

The IC has a serial digital control input and integrated charge pumps with factors of 1, 1.5 and 2, enabling it to operate with supply voltages from 2.7 V to 5.5 V. Among other features are built-in logic for avoiding thermal overload and — important for battery operation — a standby mode with current consumption typically less than 0.1 µA. In operation the IC draws around 1 mA. Maximum current for the LEDs — shared across all three LEDs — can amount to 180 mA. LED brightness is set individually in 16 logarithmic stages each, producing in total $2^{4 \times 3} = 4{,}096$ different colours. On top of this there are 16 steps of overall brilliance.

The IC operates at a clock rate of 1 MHz and with 12 pins and dimensions of just 2.4 × 3.0 × 1 mm it is extremely compact. The only external components required are four small 1 µF ceramic capacitors. A functional diagram is given in **Figure 1**. All we need to complete the circuit is a small microcontroller to provide the AAT3129 with data.

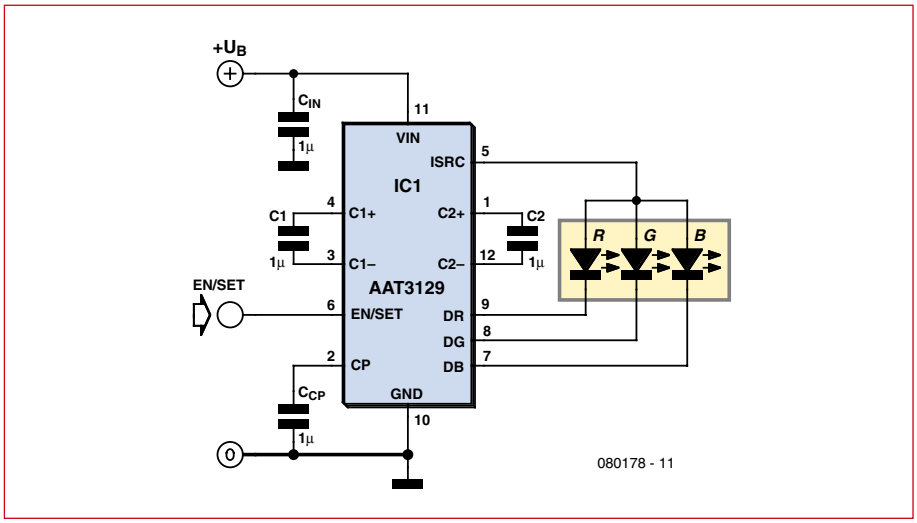


Figure 1. Block diagram of an RGB LED driver using the AAT3129.

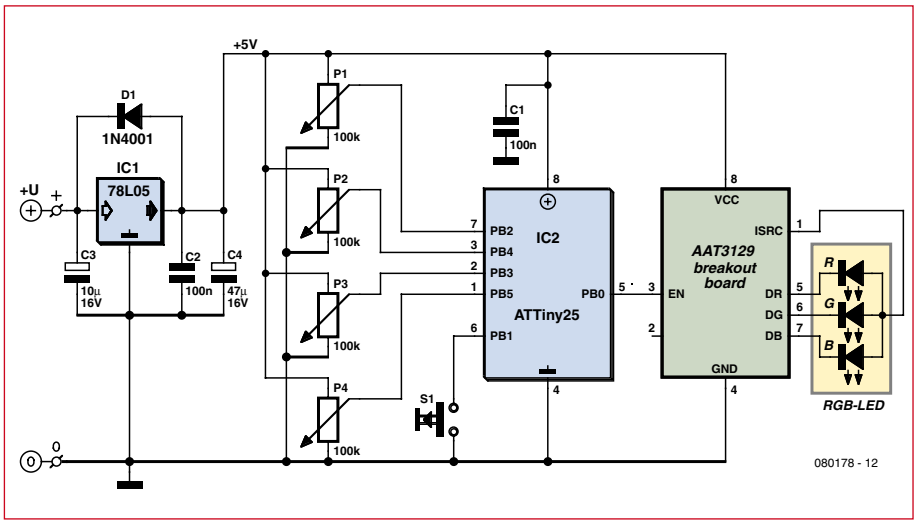


Figure 2. Control diagram with microcontroller and connector for the breakout board.

AS²Cwire

Data for the AAT3129 is presented in the Simple Serial Control (S²C) protocol, AS²Cwire [2]. The S²Cwire™ single-wire interface offers a very straightforward control technique for programmable power IC devices, using just a single wire. Data is transmitted as a series of negative-going pulses having a length of between 50 ns and 75 µs. Between pulses the level remains High for up to 500 µs. Greater values are treated as separator signals between pulse trains (see data sheet [1]). Sequences with 16 to 21 pulses are interpreted as addresses for the registers R, G, B, T (total intensity) and M (operational mode) (see **Table 1**).
The sequence that follows afterwards with 1 to 16 pulses is the actual data. To summarise, the address follows a High level of >500 µs, after which comes the data to be transmitted. Whether a value is to be executed straightaway or synchronised only once all the colour values have been defined, depends on the value placed in the M register.

Control driver

In order that we can select the colours and the overall brightness easily with rotary or slider pots, we need to use another small microcontroller with a built-in multi-channel A/D converter. This transforms the analogue potentiometer values into corresponding digital values, converts them and passes

Table 1

Register	Address	Range of values	Meaning
R (Intensity, red)	17	1-16	1: unlit 16: maximum brightness
G (Intensity, green)	18	1-16	
B (Intensity, blue)	19	1-16	
T (Overall intensity)	20	1-16	1: maximum brightness 16: darkest state
M (Operating mode)	21	1-2	1: Value is converted immediately 2: Value is converted after writing to the T registers

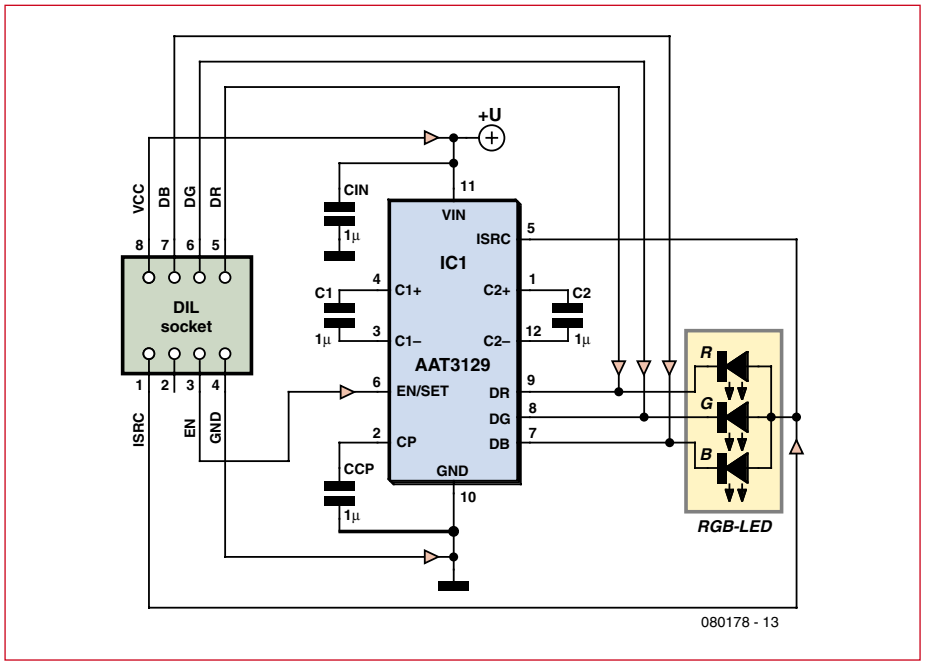


Figure 3. The circuit of the breakout board consists of just the AAT3129, four capacitors and an RGB LED if required.

the result to the driver IC. The small 8-pin ATtiny controllers from Atmel make this task a breeze. Four pins are configured as analogue inputs for R, G, B and T, whilst a changeover switch defines the operating mode. Apart from the two pins for +UB and ground, just one pin remains, the serial output that controls the AAT3129 chip. The software for the chosen microcontroller type, ATtiny 25, is covered here in a separate **inset**.

Control circuitry

The control circuitry can be seen in **Figure 2** and described in very few words. Apart from a 5-V voltage regulator this comprises a microcontroller, four 100-kΩ pots, a switch (operating with a pull-up resistor integrated in IC2), an IC socket for connecting a breakout board and finally another RGB LED if required. The breakout board is a small plug-in board equipped with the AAT3129 chip and the four capacitors mentioned previously.

The circuit is so simple that you can build it on a scrap of perfboard without difficulty. As not everybody feels comfortable with soldering the 'fine-pitch' arrangement of the pins of the AAT3129, the author hit on the idea of laying out the breakout board mentioned for the AAT3129 complete with the capacitors (and optionally an RGB LED in PLCC4 form factor) to enable it to be plugged simply into a DIL IC socket or a breadboard device or else soldered onto some 2.54 mm (.1 inch) pitch Veroboard or perfboard. The circuit of this breakout board is shown in **Figure 3**. You can download the layout files for this tiny board in KiCAD and Gerber format at the web page for this article on the Elektor website. This mini PCB does not have to be used exclusively with the microcontroller recommended here and can also be integrated into other circuits without difficulty. **Figure 4** shows a hookup corresponding to the circuit in Figure 2 in which this little PCB is placed onto perfboard along with an ATtiny25 in a DIL package.

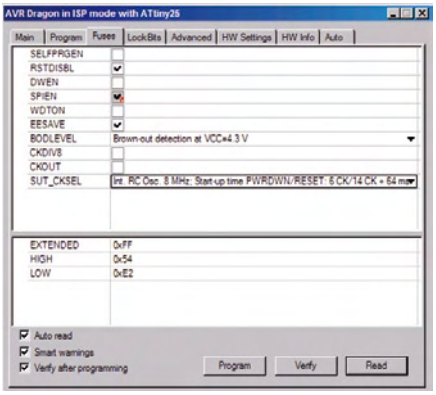


Figure 5. With the ATtiny it is vital to get the fuse settings correct, as this screenshot illustrates.

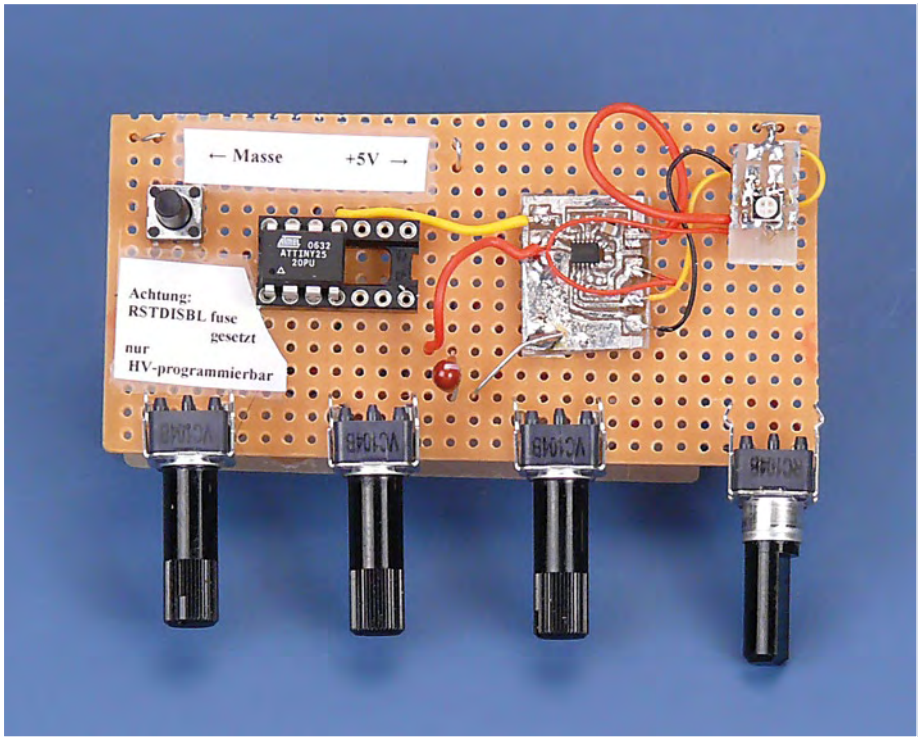


Figure 4. The author's trial set-up looks like this, with the breakout board and microcontroller built on a piece of perfboard.

Last but not least

The use of a 78L05 (IC1) means the whole circuit can be powered using any direct voltage between 7.5 and 10 V. On account of the broad supply

About the author

Fred Splittgerber has been involved with hardware-specific programming continually since the first 8-bit CPUs appeared. He works as a technical author and translator.

voltage range of the AAT3129 and ATtiny25 chips, you could also use a 3.3-V voltage regulator — or even omit the voltage regulator altogether and power the rest of the electronics direct from a stabilised 3.3 V supply. In this case the fuse for the brown-out detector needs to be matched correctly.
On the breakout board we have provided a socket for connecting an RGB LED as well as room for soldering a PLCC4 RGB LED direct. However, you should never connect two LEDs in parallel, as otherwise the necessary current splitting will not be achieved.
If the switch connected to port B1 is closed, then you will activate the colour transformation mode preset in the firmware, in which the overall brightness is set by variable resistor P4. When the switch is open circuit the RGB LED illuminates with constant brightness with the colours set with pots P1 to P3.
Source code of some sample firmware for the ATtiny25 is available for downloading free of charge from the Elektor web page [5] (see also the inset 'Software').

(080178-I)

Software

There are two important things to note in this software written in C. Firstly, the reset pin of the ATtiny25 is used as an input pin, meaning that the fuse value RSTDISBL (see Figure 5) must be defined. Once this has been done, no further SPI programming is possible. Special care is vital, as your nice new controller will turn out useless if it contains any software errors. Secondly, during colour changes the software optimises the display of uncommon greyscale hues during the transformation of one colour to another. Colour saturation is calculated according to the HSV colour model [3] and the transition between colours of low saturation is accelerated.

The serial control signal for the AAT3129 is generated using the following function:

```
void tx_pulses(uint8_t n){
  for (i=n; i>0; i--) {
    PIN_AAT=1<<AAT_BIT;
    PIN_AAT=1<<AAT_BIT;
  }
}
```

This generates 'n' pulses on Bit 'AAT_BIT' of output 'PORT_AAT'. This connection needs to present a 'high' level whenever no data is being transmitted. Here is the initialisation of the Port of the ATtiny25:

```
#define PIN_AAT   PINB
#define AAT_BIT PB0
#define PORT_AAT  PORTB
#define DDR_AAT DDRB

PORT_AAT|=1<<AAT_BIT; // Output AAT_BIT = 1
DDR_AAT|=1<<AAT_BIT;  //  AAT_BIT is Output
```

Writing to PINx toggles the polarity of the corresponding Bit of PORTx. Instead of writing `'PIN_AAT=1<<AAT_BIT'` twice we can also write:

```
PORT_AAT&=~(1<<AAT_BIT); // AAT_BIT= 0
PORT_AAT|=1<<AAT_BIT; // AAT_BIT= 1
```

With a controller clock rate of between 14 kHz and 20 MHz both methods produce the required negative-going pulses with a duration of from 50 ns to 75 µs.

For the pause that indicates the separation signal between pulse sequences you can set one of the timers or implement a delay routine. In the latter case writing the intensity '10' to the red LED looks like this:

```
#include <util\delay.h>
//…
#define CHANNEL_RED 17
//…

tx_pulses(CHANNEL_RED);// CHANNEL_RED pulse select RED register
_delay_ms(0.5);
tx_pulses(10);// Data RED register (brightness 10 from 1-16)
_delay_ms(0.5);
```

If the GCC compiler [4] is used the optimisation option '–O2' must be used.

Internet Links and Literature

[1] AAT3129 Data Sheet:
www.analogictech.com/products/digitalfiles/AAT3129.pdf

[2] AS2Cwire application notes:
www.analogictech.com/resources/applications/appnotes/AN110_S2Cwire_TLAT.pdf

[3] HSV Colour Space and Colour Space Conversion:
http://en.wikipedia.org/wiki/HSV_color_space
http://www.cambridgeincolour.com/tutorials/color-space-conversion.htm

[4] GCC Compiler for AVR:
http://winavr.sourceforge.net

[5] www.elektor.com/080178