Keywords: MAX6956, MAX6957, MAX7300, MAX7301, MAX6946, constant-current LED drivers, transition detection, driving RGB LEDs, push-pull output ports, 28 LED drive ports, max6956 programming, driving white LEDs

APPLICATION NOTE 4021

MAX6956 Programming Guide

By: Walter Chen, Principle Member of the Technical Staff, Applications
Mar 22, 2007

Abstract: This instructional guide provides detailed information on programming the MAX6956 LED display driver and I/O expander. Programming tips for similar devices (the MAX6957, MAX7300, and MAX7301) are also discussed.

Overview

Click here for an overview of the wireless components used in a typical radio transceiver.

The MAX6956 LED driver and I/O expander can provide and maintain desired constant-current levels on all of its 20 or 28 (depending on the package type) LED drive ports without the use of external current-limiting resistors. Sixteen constant-current levels (from 1.5mA to 24mA) can be applied to all ports simultaneously or selected differently for each individual port. The MAX6956 not only controls LED intensity levels, but also its port-matching accuracy makes their hues more consistent. In addition to being a constant-current LED driver, the MAX6956 allows individual ports to be used as input ports with transition-detection capability, or as push-pull output ports with a sinking current of 10mA and sourcing current of 4.5mA.

The MAX6957 is similar to the MAX6956, but it has an SPI™- instead of an I²C-compatible interface. The MAX7300 is a general-purpose I/O (GPIO) port expander that is similar to the MAX6956, except that it does not have the constant-current LED-drive capability. The MAX7301 is a GPIO port expander similar to the MAX7300, but it has an SPI- instead of an I²C-compatible interface.

Figure 1 shows a simple application example that uses the MAX6956 to drive ten white and RGB LEDs without requiring current-limiting resistors.

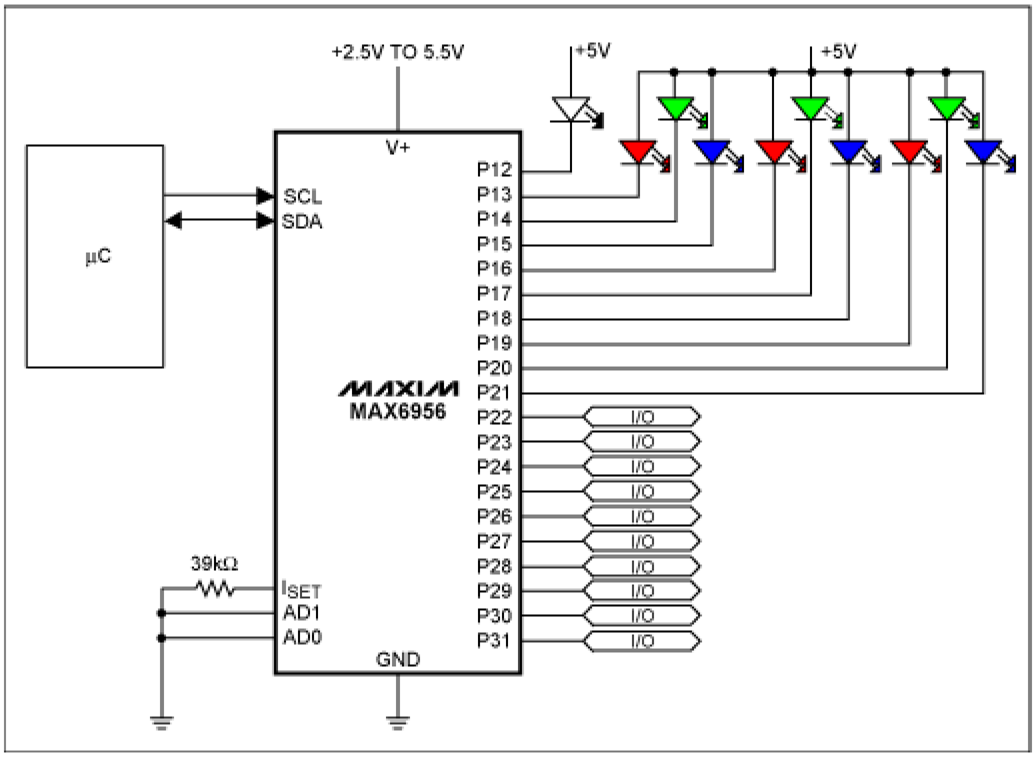

Figure 1. Application circuit for the MAX6956.

MAX6956 Configuration Registers

The detailed operation of the MAX6956 is controlled by writing to its 89 registers. In addition to the four registers (Global Current, Configuration, Transition Detection Mask, and Display Test) described in the data sheet, the MAX6956 provides three additional groups of registers:

1. Port Configuration (0x09 to 0x0F)
   There are two bits for each port in the Port Configuration register. These bits are used to define each port as an LED driver, output port, or input port with/without a pullup resistor. There are 7 Port Configuration registers corresponding to 28 ports.
2. Individual Current (0x12 to 0x1F)
   There are four bits for each port in an Individual Current register. These bits are used to define the desired constant-current level of an individual port. The minimum increment is 1/16 of the maximum level set by the external resistor connected to the I_{SET} pin.
3. Port I/O Value (0x20 to 0x5F)
   There are 29 Port I/O Value registers corresponding to each individual port (port 31 has two registers). There are 21 Port I/O Value registers that can be used to simultaneously define I/O values for a group of 8 different ports. The rest of the Port I/O Value registers can be used to simultaneously define a group of 2, 3, 4, 5, 6, or 7 different ports.

MAX6956 I²C Commands

Below are I²C commands for setting all ports to half of the maximum constant-current level of 24mA (based on a 39kΩ external resistor) at power-up. The I²C-device address of the MAX6956 is 0x80 when both AD0 and AD1 pins are connected to ground. The I2CWrite routine issues a write command to a MAX6956 register with a particular byte or a sequence of registers with multiple bytes by taking advantage of the register-address autoincrement feature.

```
I2CWrite(0x80, 0x02, 0x07);  // Set a half global constant current
```

```
I2CWrite(0x80, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00);  // Set all ports to LED drive mode
```

```
I2CWrite(0x80, 0x04, 0x01); // Set the shutdown/run bit of the configuration register
```

As a reference checkpoint the binary equivalent of the write command to set the shutdown/run bit is also listed below.

```
I2CWrite(0x80, 0x04, 0x01);

1 0 0 0 0 0 0 0         0 0 0 0 0 1 0 0         0 0 0 0 0 0 0 1
```

Below are I²C commands to set one port (P4 in this example) to the full constant-current level at power-up.

```
I2CWrite(0x80, 0x12, 0x07);  // Set a half port P4 constant current

I2CWrite(0x80, 0x09, 0xA8); // Set P4 to LED drive mode

I2CWrite(0x80, 0x04, 0x01); // Set the shutdown/run bit of the configuration register
```

All ports are set to the logic input without a pullup resistor at power-up. Only the following I²C command is needed to activate the MAX6956 to operate with all power-up defaults.

```
I2CWrite(0x80, 0x04, 0x01); // Set the shutdown/run bit of the configuration register
```

MAX6957 SPI Commands

Any register of the MAX6957 can also be written to or read from by sending a 16-bit word consisting of the register-address byte followed by the data byte through the SPI interface. The first bit of the address byte determines if it is a write (0) or a read (1) command. All 16-bit words can be put together next to each other.

Below are data bytes on the MAX6957 DIN input pin to set all ports to half of the maximum constant-current level of 24mA. These commands are similar to those for the MAX6956, except the need to specify the device address. For the SPI interface, a particular device is selected by setting the active-low CS input pin low. There is no register-address autoincrement feature for the MAX6957.

```
0x02, 0x07;     // Set a half global constant current

0x09, 0x00;     // Set ports P4 through P7 to LED drive mode

0x0A, 0x00;     // Set ports P8 through P11 to LED drive mode

0x0B, 0x00;     // Set ports P12 through P15 to LED drive mode

0x0C, 0x00;     // Set ports P16 through P19 to LED drive mode

0x0D, 0x00;     // Set ports P20 through P23 to LED drive mode

0x0E, 0x00;     // Set ports P24 through P27 to LED drive mode
```

```
0x0F, 0x00;     // Set ports P28 through P31 to LED drive mode
0x04, 0x01;     // Set the shutdown/run bit of the configuration register
```

MAX7300/MAX7301 Configuration Registers

The MAX7300 and MAX7301 do not have constant-current LED-drive capability. Detailed operations are controlled by writing to their 73 registers. In addition to the Configuration and Transition Detection Mask registers, there are two other groups of registers:

1. Port Configuration (0x09 to 0x0F)
2. Port I/O Value (0x20 to 0x5F)

MAX7300 I²C Commands

All ports are set to the logic input without a pullup resistor at power-up. Below are I²C commands for setting half of the ports, P14 through P23, as outputs at logic-level high and then bringing the chip into operation from initial shutdown. The I²C device address of the MAX7300 is 0x80 when both AD0 and AD1 pins are connected to ground.

```
I2CWrite(0x80, 0x0B, 0x5A, 0x55, 0x55);         // Set P14 through P23 to
output

I2CWrite(0x80, 0x4E, 0xFF);                     // Set P14 through P21 to
logic high

I2CWrite(0x80, 0x56, 0x03);                     // Set P22 through P23 to
logic high

I2CWrite(0x80, 0x04, 0x01);                     // Set the shutdown/run bit
```

MAX7301 SPI Commands

Any register of the MAX7301 can be written to or read from by sending a 16-bit word consisting of the register-address byte followed by the data byte through the SPI interface. The first bit of the address byte determines if it is a write (0) or a read (1) command. All 16-bit words can be put together next to each other.

Below are data bytes on the MAX7301 DIN input pin to set half of the ports, P14 through P23, as outputs at logic-level high and then bring the chip into operation from initial shutdown. These commands are similar to those for the MAX7300, except the need to specify the device address. For the SPI interface, a particular device is selected by setting the active-low CS input pin low. There is no register-address autoincrement feature for the MAX7300.

```
0x0B, 0x5A;            // Set P14 and P15 to output
0x0C, 0x55;            // Set P16 through P19 to output
0x0D, 0x55;            // Set P20 through P23 to output
0x4E, 0xFF;            // Set P14 through P21 to logic high
0x56, 0x03;            // Set P22 and P23 to logic high
0x04, 0x01;            // Set the shutdown/run bit
```

Related Parts		
MAX6946	10-Port, Constant-Current LED Driver and I/O Expander with PWM Intensity Control	Free Samples
MAX6956	2-Wire-Interfaced, 2.5V to 5.5V, 20-Port or 28-Port LED Display Driver and I/O Expander	Free Samples
MAX6957	4-Wire-Interfaced, 2.5V to 5.5V, 20-Port and 28-Port LED Display Driver and I/O Expander	Free Samples
MAX7300	2-Wire-Interfaced, 2.5V to 5.5V, 20-Port or 28-Port I/O Expander	Free Samples
MAX7301	4-Wire-Interfaced, 2.5V to 5.5V, 20-Port and 28-Port I/O Expander	Free Samples