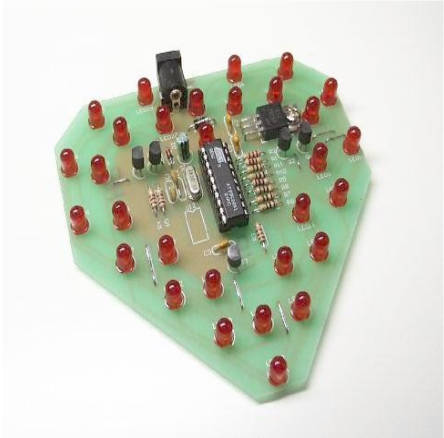
DIY Electronics
Heart of Love

Heart of Love



Here is a fun project that you can give to that "someone special".

How about *Mother's Day* or did you forget *St Valentine's Day*?

Tell someone you love them

One of the requests we received after publication of the Christmas Star (Kit 46) was "can you do different shapes?" Well, we thought a heart would be appropriate. Now you can have something different to give to that special Mum or Grandmother. Or you may be able to redeem yourself if you forgot St Valentines Day! Either way, the Heart will certainly last a lot longer than the traditional bunch of flowers!

For those who don’t like microcontroller projects, pretend the micro is a dedicated LED driver IC that just happens to have been designed to control 30 LEDs in the shape of a Heart (what a stroke of luck!). And it won’t be declared obsolete just after publication like a purpose designed IC might be!

As the design is derived from the Christmas Star, those readers who saw that article will notice that the schematic is very similar. The major differences between the Heart and the Star are in the physical layout of the LEDs and in the software.

Why use a Microcontroller?

Using a PC parallel port to control external devices is a popular approach these days but imagine the response when you present Mum with a flashing Heart attached to an umbilical cable running into the next room! No, self- contained is better. The answer is to use a small microcontroller. They are cheap and easy to use. And if the software doesn’t work first time (when does it?), you simple change the program and re-program the micro.

Also, you can easily create something using a micro that is the equivalent of many discrete logic chips. In this case, the circuit is simple enough to build on VeroBoard ™ though it is much easier to use a printed circuit board. To do the Heart in discrete logic would be a ‘challenge’ (pronounced nightmare)!

Given a few inexpensive software tools, a microcontroller such as the Atmel AT89C2051 should be just as easy to use as a handful of 4000-series CMOS chips. In my experience, the micro is usually easier and results in a more flexible design!

Another reason for using a micro is that micros are the future of electronics. While it is useful to know how to design with 4000-series logic, most new products require more than can be easily done in discrete logic. Virtually all the electronic gadgets that we now rely on (microwave ovens, mobile phones, TVs, VCRs, video cameras, fax machines, etc) use micros to perform their tasks.

So what’s in a ‘2051?

The Atmel AT89C2051 is a relatively recent derivative of Intel’s "industry standard" 8051. It includes the following features:

- 20 pin skinny-DIP plastic package
- 2k bytes of Flash program memory
- 128 bytes of RAM
- 15 programmable I/O lines
- on-chip oscillator (24MHz max)
- two 16 bit counter/timers
- six interrupt sources
- a full duplex serial port (UART)
- the I/O pins can sink 20mA for directly driving LEDs
- two I/O pins are connected to an on-chip analogue comparator

As the program memory is re-programmable Flash with 1000 erase/write cycles, the annoyance of erasing EPROMs no longer exists. Software development is now that much more convenient.

The Hardware

The heart of the hardware is, of course, the Atmel ‘2051 micro. To make it start predictably, we need a reset circuit consisting of C7. D2 forces C7 to discharge quickly when power is removed. To set how fast it thinks, we need an external crystal X1 and associated capacitors C1 and C2. Note that the specified crystal could be replaced by a ceramic resonator or crystal in the 10MHz to 12MHz range without any significant change in performance. If you use a resonator with built-in capacitors, omit C1 and C2. At 12MHz, the ‘2051 will execute an instruction every 1 or 2us.

As you can see from the schematic, the 30 LEDs are connected in an X-Y matrix. Why 30 LEDs? Engineering is full of compromises. I wanted two ‘concentric’ hearts as more interesting patterns would be possible. I achieved equal LED spacing with 16 LEDs in the outer Heart and 14 in the inner one and this "looked about right". More LEDs may have looked better but would have pushed up the cost and needed a more complicated PCB.

We can drive 30 LEDs from only 11 I/O pins using a process called multiplexing. The appropriate combination of LEDs in a column is switched on for a short time (about 2msec in this case). This process is repeated for each column in turn taking 10msec for a full cycle. Provided the multiplexing is done quickly enough, the persistence of the human eye "fills in the gaps" and we see any combination of LEDs on without any flicker. The minimum practical multiplexing frequency is about 100Hz which is the frequency used by the Heart.

The power supply uses a common 7805 three terminal regulator with bypass capacitors C4 and C5. Diode D1 provides insurance against a reverse polarity power supply. The maximum current drawn by the star is about 140mA with all LEDs on but less than about 50mA for most patterns. The maximum temperature rise of the 7805 when the star is run from a typical 9Vdc unregulated plug-pack is less than 30 degrees which is quite acceptable. It gets a little warmer when run from a 12Vdc unregulated plug-pack though it does not require a heatsink if bolted to the PCB..

Battery Operation

One question asked about the Christmas Star project was "can I run it on batteries?" Of course you can but you have to be careful depending on the actual batteries (and the voltage) used. Four alkaline cells will give about 6V which is fine if you remove the 7805 regulator and fit a wire link from it's input to it's output (outside 2 pins). Four NiCd cells are Ok too. Depending on their charge, the voltage will be between about 5.4V and 4.8V. Be very careful with SLA (Sealed Lead Acid) batteries which will charge to about 7V. Remove the regulator but use a 1N4002 diode instead of the link. Do not remove diode D1. The maximum operating supply voltage for the microcontroller is 6V (absolute max is 7V).

The Software

As with the Christmas Star, we are making the basic source code for the Heart available for free. An extended version that uses the EEPROM for storage may be available (if I get enough spare time!). Or maybe a reader would like to try. The software is written in the C language using the low cost Dunfield Development Systems Micro/C compiler. There is nothing particularly "smart" or "tricky" about the software – it was written to be easy to understand and to encourage use of small micros. Consequently, there are no interrupt routines and no use of the counter/timers, the UART or the comparator though Micro/C can make use of these resources.

The software is table driven. This means that the display patterns and sequences are determined by data stored in a table (an array of bytes). There is a simple interpreter that scans through the table to perform the specified operations. The defined byte values are listed in the following table. Note that the software for the Heart is a little smarter than for the Star – so it can do more complex pattern sequences.

Byte value or range	**Operation**
01 to 30 (0x01 to 0x1e)	Turn on LED 1 to 30
33 to 62 (0x21 to 0x3e)	Turn off LED 1 to 30
64 to 94 (0x41 to 0x5e)	Turn on LED 1 to 30 and do current delay
97 to 126 (0x61 to 0x7e)	Turn off LED 1 to 30 and do current delay
128 to 159 (0x80 to 0x9f)	Define a subroutine
160 to 195 (0xa0 to 0xbf)	Call a subroutine
196 (0xc0)	Go back to byte after loop start
197 to 207 (0xc1 to 0xcf)	Loop start, count = byte & 0x0f
208 (0xe0)	Delay (use last delay count), each count = 50ms
209 to 239 (0xe1 to 0xef)	Delay, count = byte & 0x0f, each count = 50ms
252 (0xfc)	Return from subroutine
253 (0xfd)	All LEDs on
254 (0xfe)	All LEDs off
255 (0xff)	End of table

Note that there are still quite a few undefined values so future expansion is possible.

Putting it all together

Assembly is quite straight forward. You will need a soldering iron with a fine tip, preferable temperature controlled to about 600^{o}F or 320^{o}C. Carefully check for shorts between tracks and broken tracks. Fit the lowest parts first – the wire links and resistors and diodes. Next, fit the crystal (or resonator) and the IC sockets for the micro and EEPROM (if used). Fit the transistors, capacitors and LEDs. Pay particular attention to the orientation of the LEDs – they all point the same way but they don’t work when installed backwards! Finally, install the regulator and power socket.

Do another close visual inspection looking for solder bridges especially on the transistor pads.

Connect power and check for 5Vdc (4.8V to 5.2V) from U1 pin 20 (+) to U1 pin 10 (-). If all is OK, remove power, plug in the micro and turn it on.

The hole near LED2 may be used to hang the heart. If you hardwire the power supply, you may be able to use this hole as a strain relief and hang the Heart on the power wires.

Finally, the appearance of the Heart may be enhanced by placing a piece of red cellophane over the front.

Fault finding

5Vdc not present: Check the applied power polarity – the centre pin of SK1 must be positive. Check that D1 is correctly fitted. Check tracks from SK1 via D1, the 7805 to U1 for breaks or shorts.

One LED does not work: It may be inserted backwards or it may be shorted.

One group of adjacent LEDs does not work: Check circuitry and soldering around the appropriate ‘column’ drive transistor.

Several individual LEDs do not work: Check the corresponding ‘row’ drive circuitry.

Remember, faulty components are rare, soldering problems are not so rare!

The future

The Heart is still evolving. That is part of the attraction of using a micro – it is so easy to change the behaviour by changing the software. Sometimes it is difficult to know when to stop... And what about that EEPROM? Well, an enhanced version of the Heart would read its data from the EEPROM for much longer sequences. This development will depend on available time so there are no guarantees! To check out the latest version of the software, log in to our Web site at http://www.grantronics.com.au or find us via the Silicon Chip web site.

We hope you have as much fun building the Heart and playing with the software as we did creating it. Enjoy!

Parts List

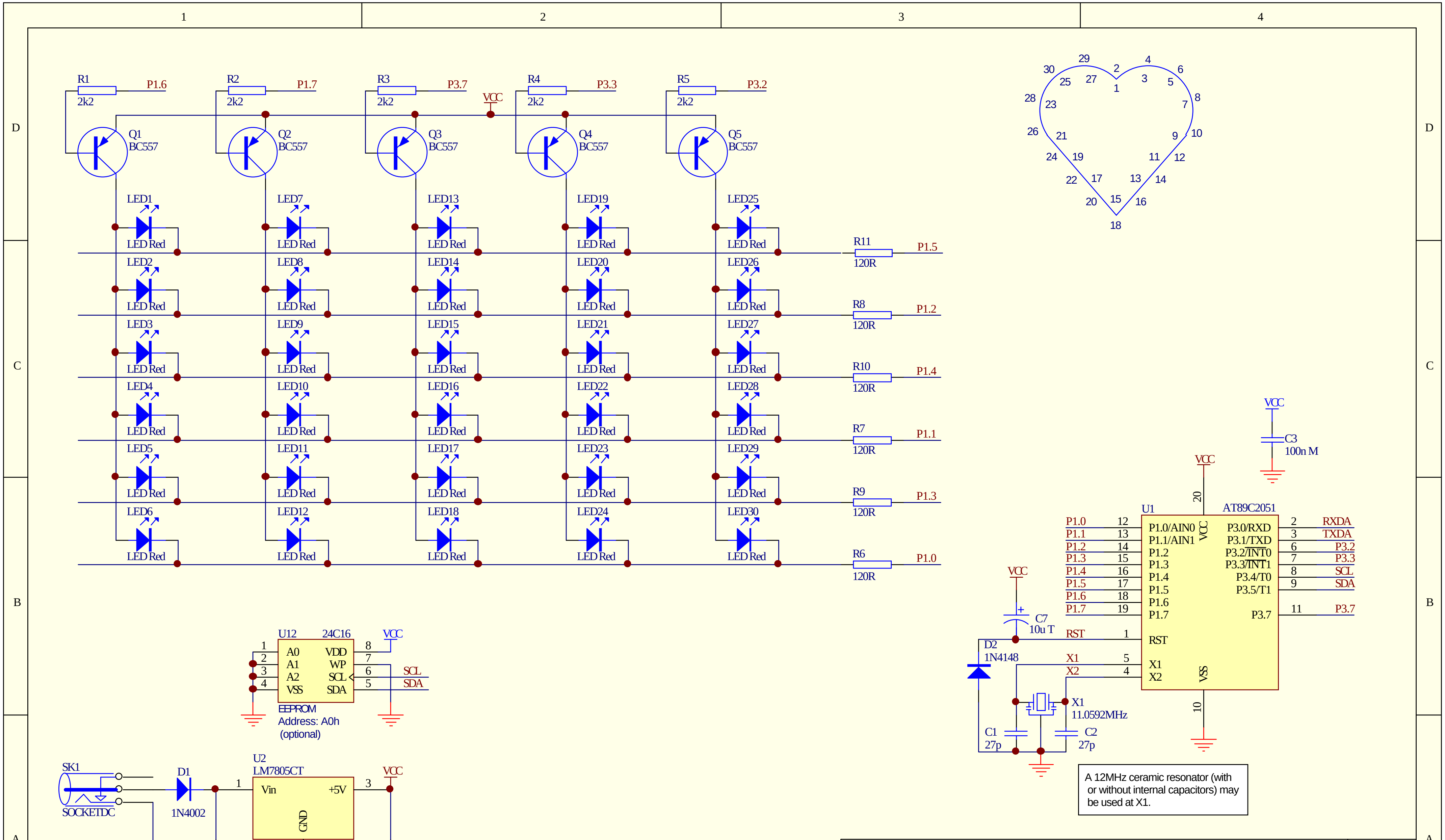
Quantity	Description
Hardware	
1	PC board type HEART
1	Power jack connector, 2.5mm (SK1)
1	11 - 12MHz crystal (X1)
1	8 pin IC socket (optional)
1	20 pin IC socket
1	9Vdc 150mA plug pack power supply
Semiconductors	
1	AT89C2051 (programmed) (U1)
1	7805 (or LM340T-5) regulator (U2)
1	24C16 EEPROM (U12) (optional, enhanced version only)
30	Super bright red LEDs (LED1-LED30)
5	BC557, BC558 or similar PNP transistor (Q1-Q5)
1	1N4004 power diode (D1)
1	1N4148 diode (D2)
Resistors (0.25W, 5% tolerance)	
5	2k2 (R1-R5)
6	120R (R6-R11)
Capacitors	
2	27p ceramic (C1-C2)
3	100n monolithic (C3-C5)
1	10uF electrolytic (C7)

Note: The LEDs should be relatively bright diffused types (about 50mcd).

To compile the software, you will need a C compiler such as the Dunfield Micro/C.

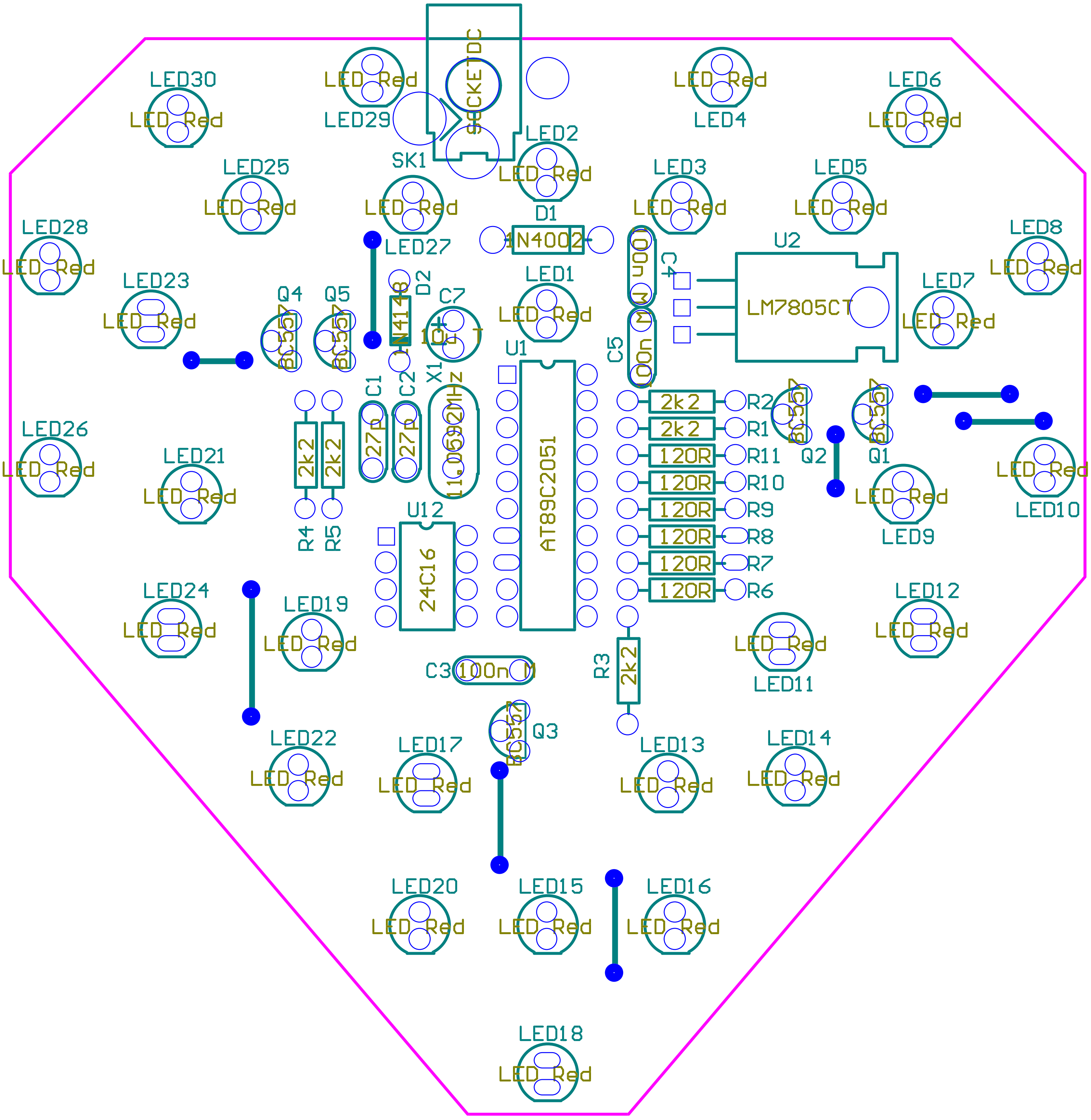
To program the AT89C2051 microcontroller, you will need a suitable programmer such as our Kit 121.

This page was last updated (american format): 08/11/2011 13:20:15



A 12MHz ceramic resonator (with or without internal capacitors) may be used at X1.

Title: Grantronics Heart		
Number: HEART	Revision: 0 Sheet: 1 of 1	***GRANTRONICS PTY LTD*** *45 Monash Street* *PO Box 275* *Wentworthville NSW 2145* *Australia.*
(02)9896-7150 info@grantronics.com.au	Drawn: L J GRANT	
Copyright (c) 1999 Grantronics Pty Ltd	Date: 9 Mar 1999	
File: E:\GRANT\DESIGN\HEART\HEART.SCH		Size: A4


LED30
LED Red
LED Red
LED29
SOCKETDC
SK1
LED Red
LED4
LED6
LED Red
LED2
LED Red
LED25
LED Red
LED Red
LED3
LED Red
LED5
LED Red
LED28
LED Red
D1
1N4002
LED27
U2
LED8
LED Red
LED23
LED Red
Q4
BC557
Q5
BC557
1N4148
D2
C7
10u
100n M
C4
LM7805CT
LED1
LED Red
LED7
LED Red
C5
100n M
U1
X1
11,0592MHz
C1
27p
C2
27p
2k2
R2
2k2
R1
120R
R11
120R
R10
120R
R9
120R
R8
120R
R7
120R
R6
BC557
Q2
BC557
Q1
LED26
LED Red
LED21
LED Red
R4
2k2
R5
2k2
AT89C2051
LED Red
LED10
LED Red
LED9
U12
24C16
LED24
LED Red
LED19
LED Red
LED Red
LED11
LED12
LED Red
C3
100n M
R3
2k2
BC557
Q3
LED22
LED Red
LED17
LED Red
LED13
LED Red
LED14
LED Red
LED20
LED Red
LED15
LED Red
LED16
LED Red
LED18
LED Red