# Senior Design Project:  Blind Transmitter


Marvin Lam
Mamadou Sall
Ramtin Malool
March 19, 2007

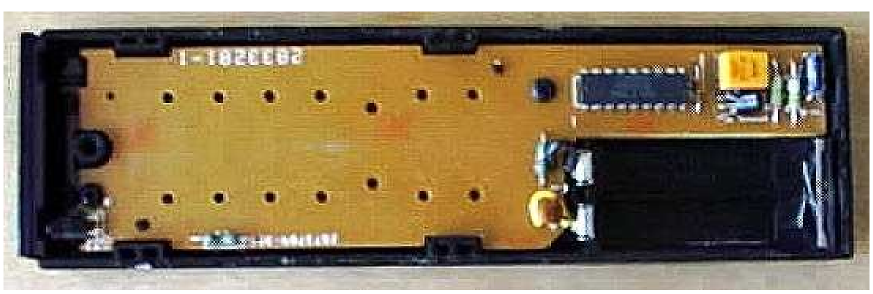
As the technology industry progresses we cannot help but to note that products are becoming both smaller and faster. It would be a waste of time and effort to not follow the industry's trend. Our group is comprised of three male engineers who all had the same goal at the start of this project; to better our everyday lives.

This device will make you laugh out loud just thinking about it. Close your eyes and picture a crowded sports bar, people packed in shoulder to shoulder, and the TV is blaring. The cute girl sitting across from you talking but you cannot hear anything with the blaring TV drowning every last breath out of the poor girl. So, you reach into your pocket and with a quick click of a button the TV turns off and you're back in the conversation with other social abnormalities to conquer. However, with our blind transmitter device that turns TV's on and off you were able to side step an unneeded social barrier. This device could be made so small that if it ever goes into mass production it can be made in a practical size that could be easily concealed in your pocket, it's possible to make this device in 1½ "x 2½" x less than ¼" thick! This control is made to only turn the TV on when it's off or turn it off when it's on. It is also possible to extend this project and make a device that can change the channel or even adjust the volume level, but for now our device only is able to turn the TV off and on. Just imagine the fun with this at a party, or the lounge at the doctor's office. The mini TV remote works on all Sony TV's. It is also possible to program this device so that it would be able to turn any TV off and on given the specific codes for that TV.

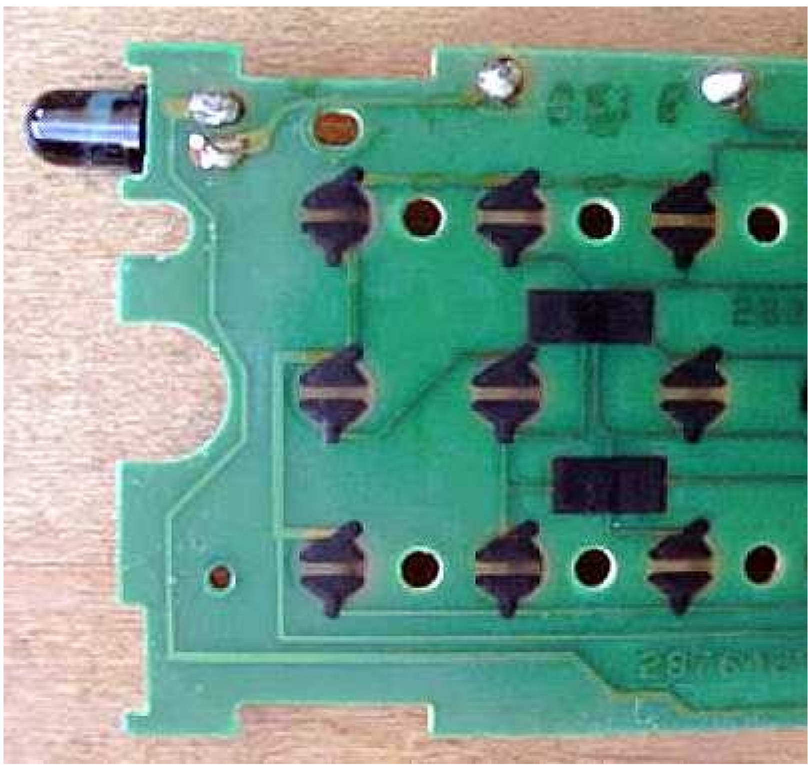
The end goal of our project is to design a microcontroller that will blindly transmit the ON/OFF signal to our desired target television. The scope of this project will extend beyond just functionality and ultimately factor in size and cost. We want to design a low power blind transmitter to fit in our consumer's pocket.

If you are like most Americans, you probably pick up a TV remote control at least once or twice a day. Let's look inside and see how they work. Here is the remote we will be dissecting today:

The remote control's job is to wait for you to press a key, and then to translate that key-press into infrared (pronounced "infra-red") light signals that are received by the TV. When you take off the back cover of the control you can see that there is really just 1 part visible: a printed circuit board that contains the electronics and the battery contacts.



The components that you see here are typical for most remotes. You can see an integrated circuit (also known as a chip) labeled "TA11835". The chip is packaged in what is known as an 18 pin Dual Inline Package, or a DIP. To the right of the chip you can see a diode, a transistor (black, with three leads), a resonator (yellow), two resistors (green) and a capacitor (dark blue). Next to the battery contacts there is a resistor (green) and a capacitor (tan disk). In this circuit, the chip can detect when a key is pressed. It then translates the key into a sequence something like Morse code, with a different sequence for each different key. The chip sends that signal out to the transistor to amplify the signal and make it stronger



At the end of the circuit board there is an infrared LED, or Light Emitting Diode. You can think of an LED as a small light bulb. Many LEDs produce visible light, but a remote's LED produces infrared light that is invisible to the human eye. It is not invisible to all eyes, however. For example, if you have a camcorder it can see the infrared light. Point your remote at the camera and push a button. You will be able to see the infrared light flashing in the viewfinder. The receptor in the TV is able to see infrared light as well.

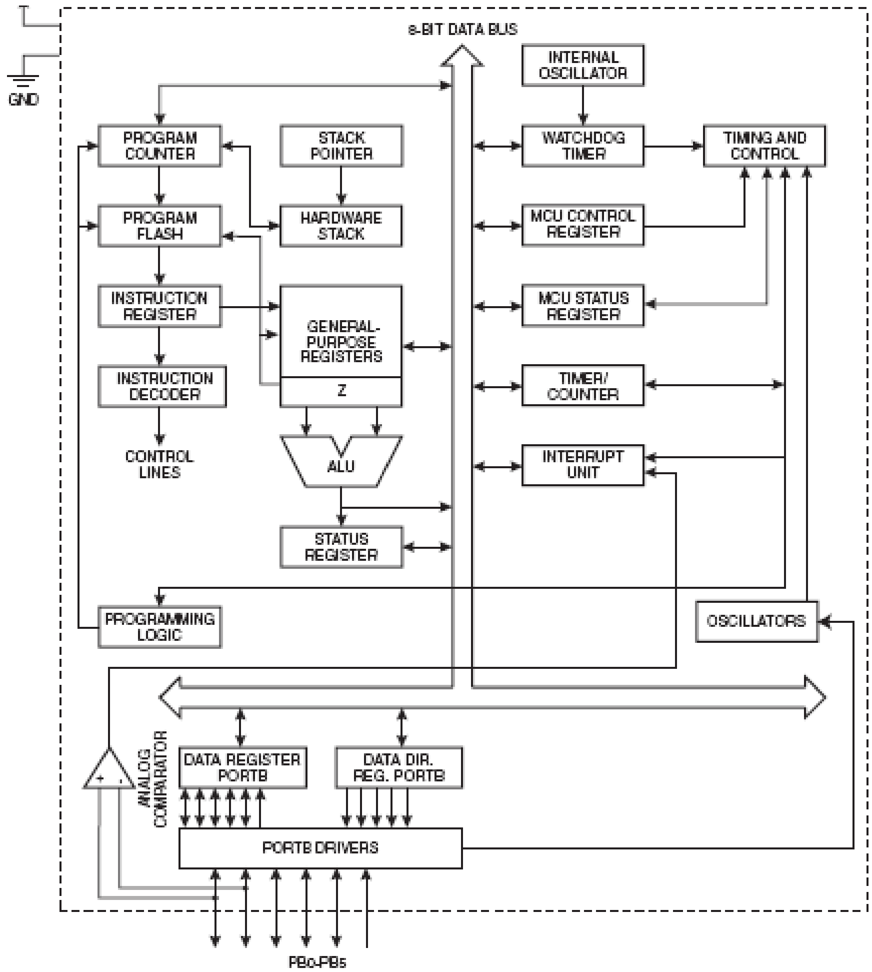
So the basic operation of the remote goes like this: You press a button. When you do that you complete a specific connection. The chip senses that connection and knows what button you pressed. It produces a Morse-code-line signal specific to that button. The transistors amplify the signal and send them to the LED, which translates the signal into infrared light. The sensor in the TV can see the infrared light and "seeing" the signal reacts appropriately

In our design we are looking for very small low power microcontroller because of the obvious practical reasons. We also use very small batteries with long mAh battery life. We also use a long range, high power infrared LED. As it was described in the paragraph above, the remote needs a microcontroller to send our signals to the TV and therefore we chose to use the Atmel ATtiny11. It is very small in size and also has internal clock along with 32 registers. The ATtiny11/12 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny11/12 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed. The AVR core combines a rich instruction set with 32 general-purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATtiny11 provides the following features: 1K bytes of

Flash, up to five general-purpose I/O lines, one input line, 32 general-purpose working registers, an 8-bit timer/counter, internal and external interrupts, programmable Watchdog Timer with internal oscillator, and two software-selectable power-saving modes.

The idle mode stops the clock while allowing the timer/counters and interrupt system to continue functioning. The power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. The wake-up or interrupt on pin change features enable the ATtiny11 to be highly responsive to external events, still featuring the lowest power consumption while in the power-down modes. The device is manufactured using Atmel's high-density nonvolatile memory technology. By combining an RISC 8-bit CPU with Flash on a monolithic chip, the Atmel ATtiny11 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

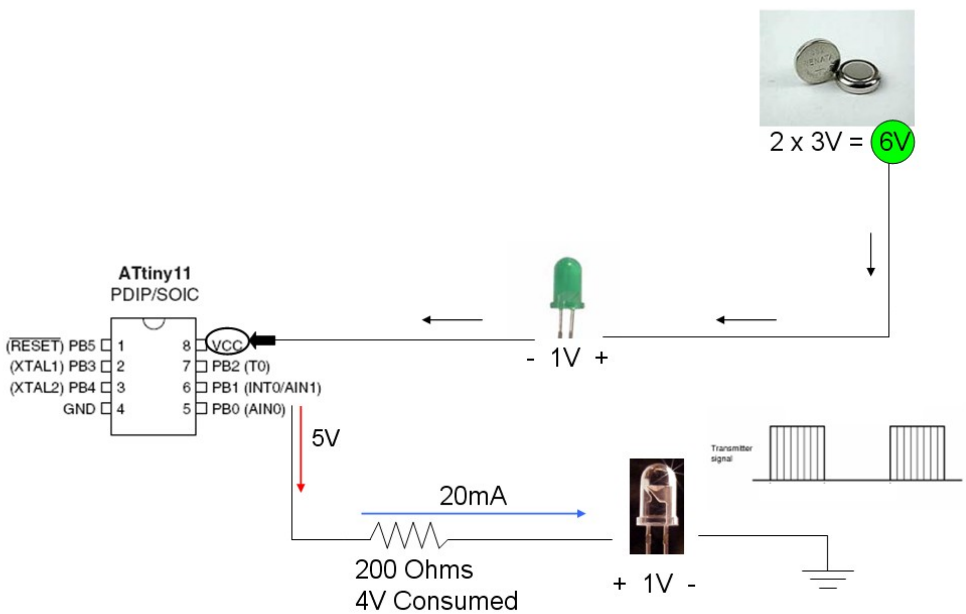
The ATtiny11 Block Diagram




■

We programmed this microcontroller using Atmel's STK 500. The AVR's supporting electronics are physically and logically separated into three distinct areas on the printed circuit board. The most colorful area is where the target AVR's reside. You can program a variety of AVR's in these color-coded sockets and then run the freshly programmed application without having to move the target AVR. Each family of AVRs has a different set of programming pins and possibly a different programming algorithm. To accommodate the program mode differences, the Atmel STK500 development board associates a color-coded set of programming header pins with similarly colored set of target AVR sockets. The

STK500 board has provisions for in-circuit serial programming (ISP), high-voltage serial programming, and high voltage parallel programming. High-voltage serial programming is used on the smaller devices that don't have enough pins to be programmed with the parallel method. The "high voltage" is actually 12 VDC and is usually applied to the RESET pin of the AVR. In addition to the AVR sockets, jumpers to include or exclude target AVR operating and reference voltages, oscillators, and crystals are also found in the color-coded area. All of the AVR socket signals are brought out to expansion headers in case you have bigger plans than the STK500 development board can handle on its own.




As you can see in the last figure, we connect our already programmed ATtiny11 to the battery source. The requirement voltage for this microcontroller is

5 volts. But the small batteries that we required in the project only come in 1.5 and 3 volts. Therefore in order to deliver 5 volts of voltage to our chip, we need to take away one extra volt that is created by putting two 3 volts batteries in series together. We do that by using a LED light that consumes one volt when voltage is applied to the end of it. The other end of the micro chip is connected to a 200 ohms resistor in series with the infrared LED. We need a current of 20mA through our infrared LED, and we achieve that by using the right size resistor (200 ohms).

Programming the microcontroller using assembly represented about 50% of the work for this project. Soldering the circuit was not very complicated, but one had to have both a circuit and the platform to program the microcontroller in order to do any simulation. The cost of microcontroller and pc boards being relatively low, we had the luxury to buy enough equipment for each of the three members. In fact it was more out of necessity than luxury since it was impossible to divide the work. We had to meet every week and work together or meet every other week with each member doing their work separately. We chose the latter one, which happened to be successful. Every single person in the group programmed their microcontroller, as well as simulated it using a SONY television. We were able to debug two out of the three circuits and finally chose one for our presentation. We successfully turned on and off any type of SONY television giving much satisfaction and pride to each of us. The infrared LED chosen allows the device to work up to 80 feet, making it very powerful and useful in public places.

The first major improvement that can be made to this project is to implement the device so that it works beyond Sony brands. The difference between Sony and any other brand is the timing at which their devices retrieve information. Sony resolves to define their start bit as a 2.4ms high bit followed by 0.6ms low.  The rest of the signal will be composed of 0's and 1's with a 0 bit defined as a 0.6ms high and a 1 bit defined as a 1.2 ms high, and are both followed by 0.6ms low. Therefore, in order to extend it to other brands, the same program could be used with the number matching the other brands being targeted.

The group encountered a couple of non-technical issues, and one of them was to decide what brand we would target in order to build our prototype. After a short debate, we agreed that Sony was the most appropriate one due to its popularity. Moreover, their protocol for the coding of televisions was more accessible than the other brands, which are more reluctant at giving it up. Another non-technical issue encountered was to decide the times at which the group was supposed to be meeting. In fact, with each member having different schedules due to work and personal commitments to other projects, we had a hard time deciding the days we want to meet. Since we didn't need the lab very often we agreed that Sundays would fit everybody's schedule and we also would meet once a month on a Friday afternoon at the lab to perform the tasks needed in each period.

The team chemistry was flawless, because first of all, we have been very good friends for over three years now, and know each other's habits. We for sure have had our moments, due to each others drive for perfection and excellence, but it was strictly on the professional aspect, and none of the arguments were ever personal. The communication was great and it was a joy to actually work with motivated but yet very relaxed individuals who took the project very seriously.

The budget did not exceed one hundred and fifty dollars. It allowed us to buy the microcontrollers, LEDs, and the platform in which they are programmed, because like we said earlier, each member had to build their own circuit. The ATtiny11 being a cheap microcontroller, we were certainly in good shape financially. Based on the success we've had in making the devices work, one can say that it was a cost effective project.

As our projected winded down and we were able to look at our work in retrospect, the group agreed that we chose a suitable project that had the right motivation. Sure, there were times we wanted to quit but we believed that we were very capable of finishing and wanted to present a product that works. The overall feeling for this course and project was unanimous. Not only was it a great learning experience on a technical level, but we were also able to develop teamwork skills as well as coping with the nerve racking experience of a public presentation. We have all agreed that this project has helped us develop integral parts of our professional career.

## Appendix

1. All microcontroller parts and programs provided by, www.atmel.com.
2. Batteries and LED's store bought from, www.radioshack.com.