The creation of the THEREPING

By Vern Graner

Introduction

The "Thereping" is a digital musical instrument that plays sounds based on a combination of the position of your hand and some pushbutton switches. It uses a combination of a sonar sensor and a microcontroller to allow the user to play interesting melodies without requiring any musical skills. When joined together with a "Thereclock" sync unit, multiple Therepings can all play together and create interesting music, again without requiring any musical expertise on the part of the players.

In the beginning

The quest for a new and interesting musical instrument began when The Robot Group was selected to participate in the First Night Austin celebration with a proposal they entitled "The Robot Theremin Band". The idea was to use theremins (see sidebar: "Whats a Theremin?) and robots together to create an experimental, musical spectacle. However, the Theremin is a difficult instrument to play *well* as it requires a high degree of accurate physical control (i.e. hand/body position) as well as a good knowledge of music theory. As far as expertise is concerned, it has much in common with a violin in that if played well, it can be beautiful. If *not* played well, it can be... unpleasant at best.

The First Night project was being discussed and advanced for the most part through The Robot Group's mailing list. After reading a description of the proposal, I posted that "a group of technically minded folks, with no musical training, trying to play Theremins would sound roughly like someone attempting to murder a large number of cats with a mallet". Since this might be more apt to *repel* people than attract them to our presentation, I suggested that we might consider crafting some sort of electronic musical instrument instead. The topic received little more attention as the First Night performance seemed rather far away ("seemed" being the key word!).

What's a Theremin?

The theremin or thereminvox is one of the earliest fully electronic musical instruments. Invented in 1919 by Russian Léon Theremin, the theremin is unique in that it requires no physical contact in order to produce music and was, in fact, the first musical instrument designed to be played without being touched. The instrument consists of a box with two projecting antennas around which the user moves his or her hands to play. To control the theremin, the musician stands in front of the instrument and moves his or her hands in the proximity of two metal antennas, the distance from the antennas determining frequency (pitch) and amplitude (volume). Small movements of the hands can create a tremolo or vibrato effect. Typically the right hand controls the pitch and the left hand is used for the volume, although some play left-handed. Based on the principle of heterodyning oscillators, the theremin generates an audio signal by combining two different but very high frequency radio signals. The capacitance of the human body close to the antennas causes pitch changes in the audio signal, in much the same way that a person moving about a room can affect television or radio reception. By changing the position of the hands relative to the vertical antenna, a performer can control the frequency of the output signal. Similarly, the amplitude of the signal can be affected by altering the hand's proximity to the looped antenna. The information above is excerpted from "Wikipedia", located at the following URL: http://www.wikipedia.org

Ahhhhhhh... FREQ OUT!

In mid November 2005, at a meeting held at my house (Figure 9), Don Colbath (one of the group members who actually owns a *real* Theremin) was toying with a small test system I was using to demonstrate the Parallax Basic Stamp II connected to a Parallax PING)))™ sonar sensor (Figure 3). The test system was running a program that would show the distance of an object from the PING)))™ sensor on an LCD display. Don remarked that if the distance reading could somehow be sent to an audio oscillator, it might be possible to make a sound similar to a Theremin! Intrigued, I wrote some very simple PBASIC code that would measure the distance reported by the sonar sensor and store it in a variable. Then, rather than try to coble together an oscillator, I just stuffed that value into the PBASIC command used to create sounds (the aptly named "FREQOUT") to have the Basic Stamp itself produce sound directly as shown here:

Figure 1: Gray Mack and the Prototype Thereping

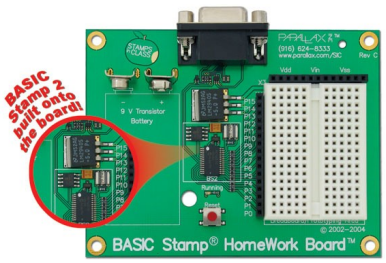
```
PING     PIN 4    ' Parallax PING)))™
SPEAKER  PIN 9    ' Speaker
SAMPLE   VAR Word ' Store result

AGAIN:
  PULSOUT PING, 5
  PULSIN PING, 1, sample
  FREQOUT SPEAKER, 100, sample
GOTO AGAIN
```

When we ran the above code, the unit began to play sounds at a pitch in proportion to the distance of your hand to the PING)))™ sensor! We had a basic proof of concept for a new musical instrument. The "Thereping" was born!

Parts is Parts

Now that we had a proof of concept, it was time to order in the components so we could fabricate a bunch of instruments. I spoke with Denise Scioli, a member of the The Robot Group who had been in close contact with the First Night folks and she agreed to contact Parallax and order in all the parts we needed. Since the prototype instrument was based around the Basic Stamp II microcontroller from Parallax, and we planned to build 6 instruments, we decided on the Parallax "Homework Board" (Figure 2) to act as the base as it is available at a good price since it's sold in "10 packs". Denise placed an order and soon we were in possession of ten "Homework Boards" and ten PING)))™™ sonar sensors. In preparation for the project, Denise had created and decorated some wooden "cutouts" (Figure 8) in fanciful shapes of rocket ships and robots that were to be used as a base to hold the homework boards and provide a "familiar' playing paradigm (i.e. "guitar like"). I attached one of the homework boards and a PING)))™ sensor to one of the boards to get a feel for how the instrument would look and feel (Figure 1).

Figure 2: The Parallax Homework Board



Since we were planning on making a "band" of instruments, we decided the best approach would be to buy identical components. Subsequently we chose to purchase a "10 pack" of Parallax Homework boards. These are self-contained Basic Stamp 2 microcontrollers with a built in Breadboard and DB9 serial interface for programming. The Homework Boards are only sold in a ten or twenty piece pack, but this would give us enough units to have six instruments, one "master sync" clock and a few "spares" to use for experimentation and as backups in the event of a component failure. At quantity ten, the per-board price drops to only $40.00.

Refining the idea

Though we had one prototype circuit and mount for our fledgling Thereping, and this prototype would make sound, it was not necessarily a *pleasant* sound. Also, like an actual

Theremin, it required some amount of skill in music and physical control to create consistent musical notes and/or melodies. More importantly, if we built a number of these units they would suffer the same weakness from which a group of "real" Theremins would have suffered (i.e. the "malleted felines" syndrome). We needed the ability to place musically "unskilled" folks into the role of "musician". We also had a little under 6 weeks to have a complete "Band" ready to perform. I decided to break the problem into pieces and solve the problem one piece at a time.

What can go wrong?

For our purposes, I decided that there are two *fundamental* things that you can do "wrong" when playing an instrument:

1) You could play an incorrect note
2) You could play a note at the wrong time

I decided to attack the "incorrect note" issue first. I knew there were scales of notes where any of the notes in the scale, when played together, would sound "correct". I decided to use a "blues" scale in the key of C for my experiment. On the prototype instrument, I created code that would instruct the PING)))™ sensor to fetch a distance reading, and then would check the returned reading against a range of values to determine what note should be "played". The first code looked like this:

```
AGAIN:
  PULSOUT PING, 5
  PULSIN PING, 1, sample
' C blues scale      C - Eb - F - Gb - G - Bb - C
IF SAMPLE1 > 1000 AND SAMPLE1<1047 THEN FREQOUT SPEAKER,100,1047 ' C
IF SAMPLE1 > 1047 AND SAMPLE1<1245 THEN FREQOUT SPEAKER,100,1245 ' Eb
IF SAMPLE1 > 1245 AND SAMPLE1<1396 THEN FREQOUT SPEAKER,100,1396 ' F
IF SAMPLE1 > 1396 AND SAMPLE1<1480 THEN FREQOUT SPEAKER,100,1480 ' Gb
IF SAMPLE1 > 1480 AND SAMPLE1<1568 THEN FREQOUT SPEAKER,100,1568 ' G
IF SAMPLE1 > 1586 AND SAMPLE1<1864 THEN FREQOUT SPEAKER,100,1864 ' A#/Bb
IF SAMPLE1 > 1864 AND SAMPLE1<1976 THEN FREQOUT SPEAKER,100,1976 ' B
IF SAMPLE1 > 1976 AND SAMPLE1<2093 THEN FREQOUT SPEAKER,100,2093 ' C7
GOTO AGAIN
```

Figure 3: PING)))™ Sensor

The Parallax PING)))™ sensor uses ultrasonic sound waves to determine the distance to an object and returns a value in milliseconds that is quite accurate. In addition it is very easy to use and connect since it only requires three connections, 5v+, GND and a single TTL level pin for both sending and receiving data. The Parallax website offers the these units in a "five pack" so it was possible to leverage quantity purchase prices and buy two 5-packs of PING)))™ sensors to match the one ten pack of Homework boards. This brought the price per sensor down to about $20.00.

Though not very *efficient*, the above code did brute-force determine the hand distance above the sonar sensor and then make it so the instrument would only play corresponding notes that would be "valid" in the blues scale. When this code is run, you can move your hand above the sensor and the note that corresponds with your hand position is played. However, the timing between the notes is fixed at 100ms (roughly 16th notes at 120bps) and your hand position is absolute (i.e you cannot change the distance your hand must travel to create specific notes).

At the next meeting of The Robot Group, I showed the code to Eric Lundquist, another long-time group member (and a programmer by trade), and he had some great ideas on how to optimize the program. Together we refined the code in a number of ways to make the instrument more efficient and flexible. First, it was decided to break the space above the PING)))™ sensor into discrete "zones' that could be "calibrated" to make the sensing area adjustable and allow the notes to be scaled to different octaves or pitches. This was accomplished by determining the highest and lowest PING)))™ sensor readings in milliseconds acquired while comfortably moving our hand over the

sensor (from a bit less than 1” to about 8”) and setting those values into constants as shown here:

```
LOWPos  CON 100    ' MS Value for closest note
HIGHPos CON 800    ' MS Value for furthest note
```

Then we set a constant to hold the Number of Notes into which we wanted the range divided:

```
NNotes CON 7 ' Number of notes in the scale
```

This divided the airspace above the sensor into seven 1-inch zones. Now, you could tell in which “zone” the user's hand was detected by using this simple formula:

```
Zone = (HighPos-LowPos) / NNotes
```

In order to make the code easier to read and more intuitive, I used the PBASIC “CON” command to create an “alias” of the Hz values for each note in a 12-note chromatic scale. This alias would reflect the note's “name”, it's modifier (sharp or flat), and the octave. The naming convention I chose was:
<note letter><sharp/natural><octave>
For example, the notes from C6 through C7 would look like this:

```
Cn6  CON 1047
Cs6  CON 1109
Dn6  CON 1175
Ds6  CON 1245
En6  CON 1319
Fn6  CON 1397
Fs6  CON 1480
Gn6  CON 1568
Gs6  CON 1661
An6  CON 1760
As6  CON 1865
Bn6  CON 1976
Cn7  CON 2093
```

Figure 4 Thereclock & Therepings w/Drum Set

Though I did consider using the Basic Stamp itself to calculate the Hz values for each note in real time, it seemed like a lot more work and I was afraid doing so would require additional processing power that we might need later for other purposes. Subsequently I just pre-calculated the values for each note. Next, the original “brute force” method of determining the appropriate note to play was discarded in favor of a more elegant approach using the PBASIC “LOOKUP” command to pick a value from a range. The new completed Thereping program worked like this:

1. Check the distance to the hand above the PING sensor

```
PULSOUT PING, 5              ' Send a ping out
PULSIN  PING, 1, sample      ' store response in "sample"
```

2. Determine in which “zone” the hand was located, and store it in the “Note” variable

```
Note = (SAMPLE - LowPos) / Zone
```

3. Use the Note number to LOOKUP the correct note frequency and store that value in the FREQ variable

```
' C blues scale      C - Eb - F - Gb - G - Bb - C
LOOKUP NOTE, [Cn6, Eb6, Fn6, Gb6, Gn6, Bb6, Cn7], FREQ
```

4. Play the note with the FREQOUT command.

```
FREQOUT Speaker,100,FREQ
```

After step 4, just go around in a loop and do it all again. Using this new code, we had an instrument that would efficiently play only notes that would be correct in a specific scale. However, at this point we had no control of the note's duration or of its timing in relation to other instruments or an external tempo.

Plays well with others

Since the point of the project was to have everyone play *together*, we needed to attack the second thing that could "go wrong", namely the "note at the wrong time" issue. This boils down to creating a "SYNC" or central clock source. I theorized I could simply program each of the Therepings to look for some "central clock" source before playing a note, thus all the instruments would synchronize.

Originally, I considered using an LED IR beacon as a sync source. The design consisted of a tall tower (dubbed a "tempo tower") that would allow IR LEDs to be pointed downward to the stage where the instruments would be during a performance. Since our stage was to be a trailer in a parade, and a small outdoor pavilion, this solution at first appeared to be feasible. However, we discovered from the schedule that we were slated to play on the pavilion in the daytime when natural UV/IR light would drown any sync signal from the IR LED sync source. It was also pointed out that the musicians would need an uninterrupted line of sight to the "tempo tower" if continuous sound was to be maintained. If the players were to move about, their bodies might block the Therepings "view" of the tower. With these problems, it became clear we needed to rethink our sync.

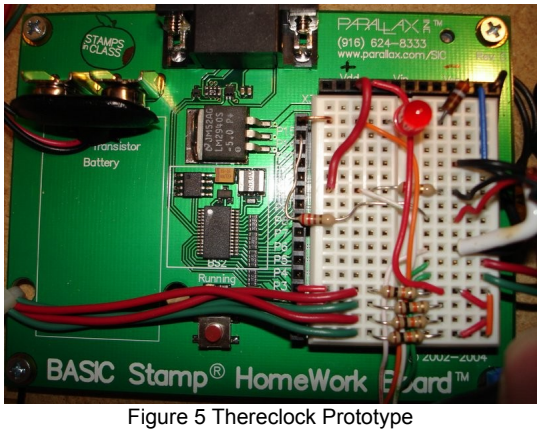


Figure 5 Thereclock Prototype

Where a clock? Thereclock!

Though the benefits of a "wireless" form of sync signal would be large (i.e. no wires to tangle, greater mobility for the musicians etc.), the downside would be that each instrument would need to contain an on-board power source, speaker system and associated audio amplifier (or even more complicated, their own radio transmitter!). Also, in order to be heard in a parade atmosphere, the Therepings would have to be equipped with some pretty beefy amps/speakers that would require some equally hefty batteries. In the end, the additional costs and associated construction time pretty much ruled out going wireless.

Since we had planned to have a central "P.A." system it made sense to simply fall back to a "hard wired" approach. I created an experimental "master clock" system (dubbed the "Thereclock") using one of the Parallax Homework Boards and some very simple code to toggle PIN 15 of the board HIGH then LOW. The code looked like this:

```
AGAIN:
  HIGH 15
  PAUSE 250
  LOW 15
  PAUSE 250
GOTO AGAIN
```

I added an LED to PIN 15 on the breadboard of the Thereclock board so I could verify the code was indeed toggling the output, then I connected a jumper from PIN 15 on the Thereclock stamp to pin 15 on the Thereping prototype (Figure 5). Note that they were sharing the same power supply so already had a common ground so I did not need to add a ground line. Once I had this complete, I modified the Thereping code to include a pin definition for "SYNC" and then added this new line of code just above the existing sound-producing command as shown here:

```
SyncWait:
IF SYNC = 0 THEN SyncWait

FREQOUT Speaker,100,FREQ
```

I powered up the Thereping unit, placed my hand over the PING sensor, and the notes began to fire off in precise accordance with the blinking of the LED on the Thereclock! Houston, we have SYNC!

Play it again... And again... and..

It was now mid December of 2005 and I had had the prototype Thereclock and Thereping circuit on my workbench for a while now as I experimented with them. I began to notice it was a bit fatiguing to hear the same exact note *durations* without any *variation*. Though it was interesting to hear for a short period of time, it occurred to me that a group of these instruments would sound very similar and it might not be possible to determine which instrument was playing. Worse yet, even if you could, the sound produced would tend to be mostly redundant. I thought it would add interest to the performance if the performer could choose the duration of the notes. For example, if the performer could choose to play 1/8th notes for a while, then play 1/4 notes, then switch to 1/16th notes whenever they chose, this would help make for a more expressive instrument. In order to check for the selection of the performer, some buttons would have to be added to the Thereping circuit, so I added PB1through PB3 to the prototype (Figure 6). I modified the code to check the buttons and alter the number of milliseconds the note would be played. When I did this, I immediately noticed that what resulted was only the ability to change the note from "filling" the entire 1/4 note interval, or to playing an 1/8th note followed by an 1/8th rest! I was making the note *duration* shorter, but I

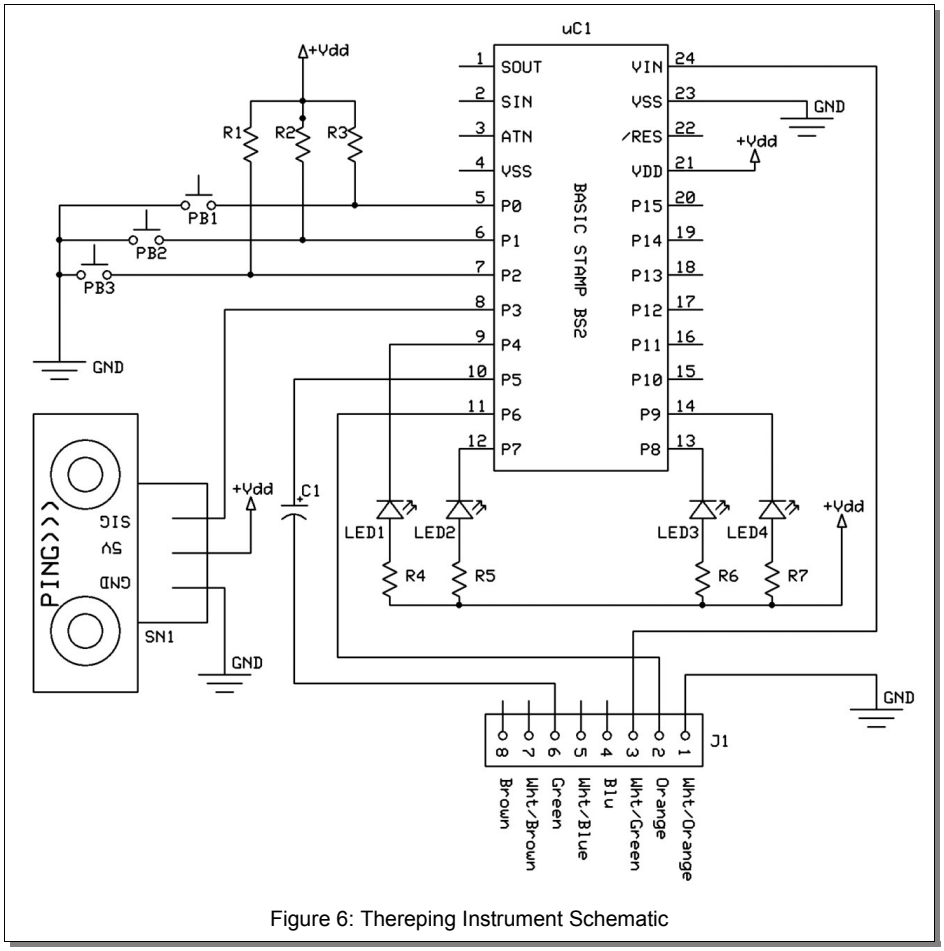


Figure 6: Thereping Instrument Schematic

was no longer filling the entire clock cycle. For those of you familiar with musical terminology, the notes just started to sound *staccato*). So, it was clear that in order to play synchronized 1/8th notes, I would have to increase the clock frequency to at least 1/8th notes. In fact, In order to allow 16th notes as well, I would have to change the clock frequency to 16th notes! When I started looking at how to do this, I discovered that I would have to have some type of mechanism to determine "where I was" in the measure in order to insure that all the instruments could start and stop together on the first "beat" of a measure. I would also need to determine the "1st" clock cycle in order to accurately divide the clock cycles back down to 1/8th and 1/4 notes when they were selected.

To illustrate this problem, imagine the clock is pulsing 16 times per 1/4 note. How do we determine *which* of those pulses is the *beginning* of the 1/4 note? This was quickly getting rather complicated, and we were also running out of time. I decided that since I knew the interval of the clock pulses coming from the Thereclock, I would be able to trigger multiple notes on the Thereping at the onset of a clock pulse before looking to the sync input for a new clock pulse.

For an 1/8th note, I altered the code to play a note *two times* with a duration that was ½ the clock frequency before returning to wait for a new clock sync pulse. For the 16th note, it would play *four* notes at ¼ the clock frequency (see flow chart in figure 7). Though not a perfect solution, this did allow me to have the performer decide on the duration of notes and led to the Therepings being able to play distinctly different sounding musical themes.

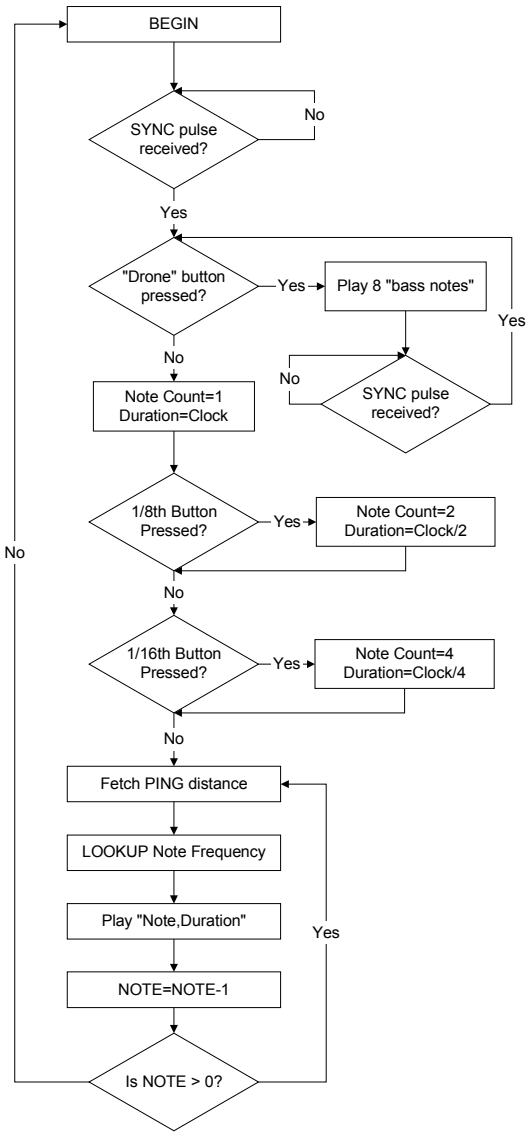


Figure 7: Logic flowchart for Thereping

Droning on and on...

Now that we had the ability to play a fairly intricate "solo" on the instrument, we were again faced with the problem that many instruments all playing similar sounds would tend to blend into just so much noise. Being exposed to modern American musical culture, most people expect certain things from "music". They expect a "rhythm section" (i.e. bass & drums), an accompaniment of some type (i.e. piano, guitar) and a discernible "lead" melody. At this point all I had was a group of instruments that could play this "lead" part. In order to create a more appealing sound, I decided that when the player of the instrument was "done" playing their "solo" part, they should be able to drop into the background and accompany or just enjoy what others were

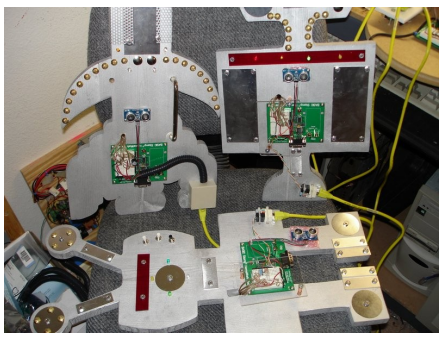


Figure 8: Three Therepings

playing. Since we were in the same key throughout the song, I knew there were certain notes that could be played that could act as a foundation to the music. In researching this problem, I came across an interesting web site on the design behind the bagpipes. The bagpipe is an interesting instrument in that it has three "horns" (referred to as "drones") that play a consistent chord of notes throughout every song. It occurred to me that I could create a sort of "drone" for the instrument using one of the buttons to tell the microcontroller that it should simply play a preset pattern of notes and ignore the sonar sensor altogether. This would let a player rest their hand, dance, look around or listen to others playing. This would also allow the instruments to play something closer to what most people expect to hear in modern music. I added code that would detect the "drone" button, and would then branch to a routine in the code that would simply repeat a series of notes in a bass line. The "drone" addition to the code is shown here:

```
DO WHILE BTN1=0  ' Check for "drone" button and play drone sound if pressed
  SyncWait1:
  BUTTON SYNC, 1, 255, 0, BtnWrk, 0, SyncWait1 ' Wait for a sync pulse

  Duration = Clock
  FREQOUT Speaker,Duration,Cn4/2, Cn4 'Play the note!
  PAUSE offset
  Duration = Clock
  FREQOUT Speaker,Duration,Cn4/2, Cn4 'Play the note!
  PAUSE offset
  Duration = Clock
  FREQOUT Speaker,Duration,Cn4/2, Cn4 'Play the note!
  PAUSE offset
  Duration = Clock
  FREQOUT Speaker,Duration,Cn4/2, Cn4 'Play the note!
  PAUSE offset
  Duration = Clock
  FREQOUT Speaker,Duration,Cn4/2, Cn4 'Play the note!
  PAUSE offset
  Duration = Clock
  FREQOUT Speaker,Duration,Cn4/2, Cn4 'Play the note!
  PAUSE offset
  Duration = Clock
  FREQOUT Speaker,Duration,Cn4/2, Cn4 'Play the note!
  PAUSE offset
  Duration = Clock
  FREQOUT Speaker,Duration,As4/4, As4/2 'Play the note!
  LOOP
```

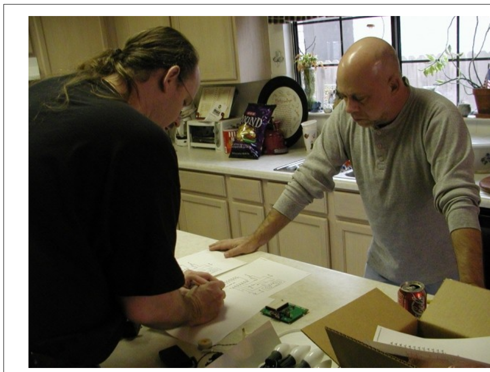


Figure 9: Vern Graner and Mike Scioli work on Thereping Prototype

One thing to notice is that the note frequency values are divided by 2 in order to drop the frequency by an octave (i.e. Cn4/2). This altered the "voice" of the instrument to more closely resemble a "bass guitar", thus creating a "driving" bass line to accompany the other players who would be "soloing". I created a couple of variations on this "drone" part in order to "change up" the sound and loaded the alternate code into two of the five Therepings. I wanted as much variety in the final song as we could get. You may also notice this is a pretty "sloppy" way to accomplish this task. From a programming perspective, it would be much tidier to place the FREQOUT commands in a loop. However, we were running out of time and most of the coding was done with an eye towards illustrating the function and allowing quick changes rather than optimizing form. Remember this if you decide to build your own Thereping, there are LOTS of opportunities for improvement!

Getting Wired

Since we had a proof of concept for the syncing of multiple instruments using wire, I now had to look at finding wiring methods to use that would easily and robustly link all the instruments to a central clock. It just so happened that I had quite a few Ethernet cables and RJ45 “keystone” jacks laying about so I decided to try using regular CAT-5 cable to connect the Thereclock unit to the Thereping instruments. The four pairs in the CAT-5 cable would allow me to provide a path for all the needed signals as shown here:

Pin	*Color*	*Use*
1	White/Orange	GND
2	Orange	SYNC
3	White/Green	V+
4	Blue	N.C.
5	White/Blue	N.C.
6	Green	AUDIO
7	White/Brown	N.C.
8	Brown	N.C.

Also, CAT-5 cables are easy to come by in many lengths and colors, and there would be enough wires to allow for future expansion if the need arose. The final schematic for the Thereping shows the CAT-5 connector as the sole I/O point for the instrument providing everything that is needed to operate it. As

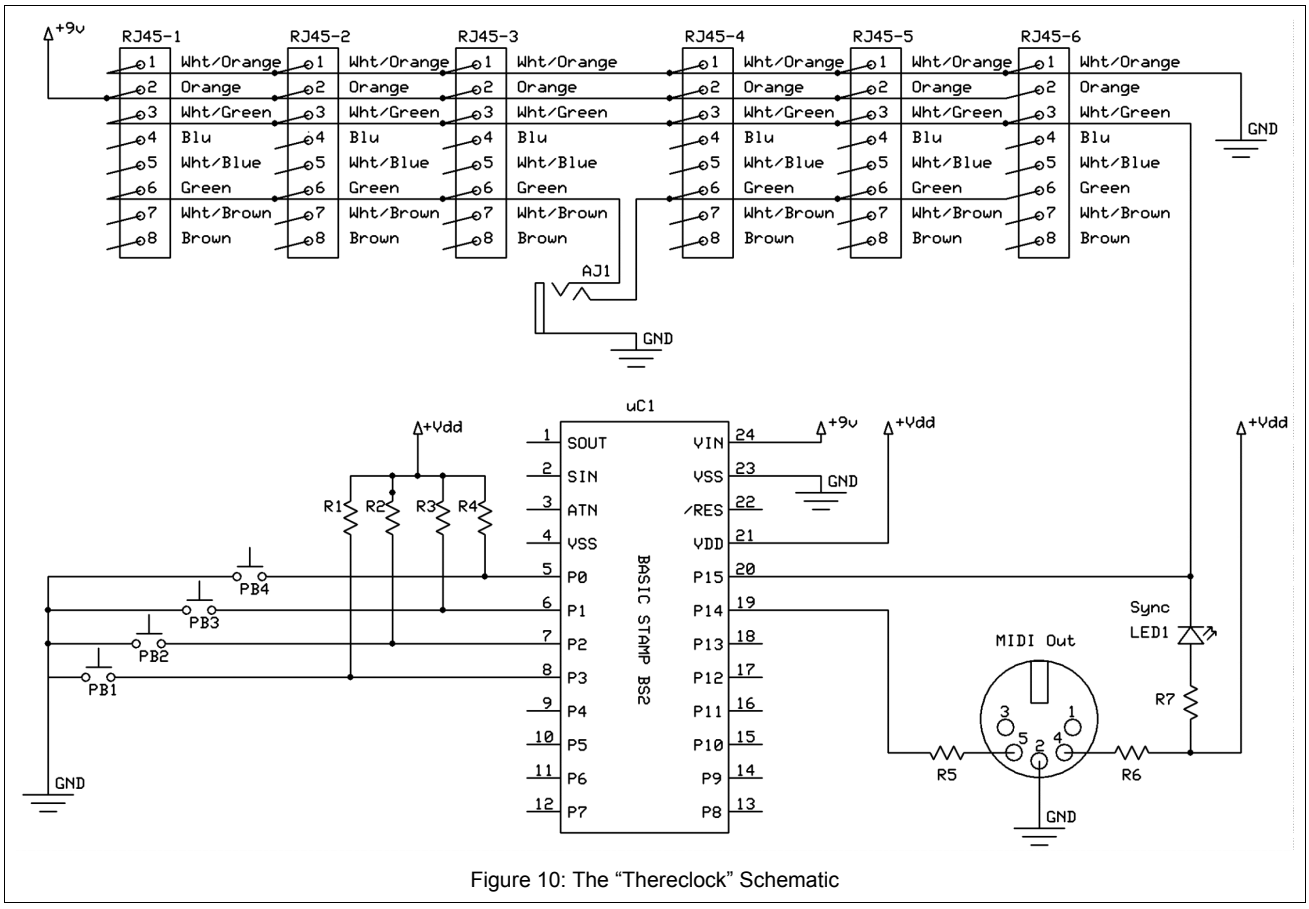


Figure 10: The “Thereclock” Schematic

part of our objective was to provide a “spectacle”, we added some LEDS to each Thereping not only to make for interesting visual effects, but also to act as an indicator that the unit was successfully receiving SYNC from the Thereclock and to be used for diagnostics and development. The final schematic for the Thereping is shown in (figure 6).

Gimme a BEAT!

Ok, we were now near to the end of December and the Thereping had the ability to play “lead” or “solo” parts, an accompaniment part via a “bass line”, and we had a way to keep all the instruments in sync. The only thing that was missing in my mind was a BEAT. Since this was designed to be used in a parade, it seemed fitting to have a rhythm playing that would provide the background for the performance. I did some research on sending MIDI from the Basic Stamp and found it was trivial to create the MIDI output hardware connection. All that was required were two resistors and one IO pin from the stamp. I took the prototype Thereclock and added the resistors and then cut up an old MIDI cable to connect to the bread board (Figure 5). I plugged the MIDI cable into the MIDI IN of the Yamaha DTXpressII sound module on my drum kit, and then went in search of some example code. It turns out that to make a Basic Stamp send a simple MIDI command is surprisingly straight forward. The following code example plays a constant 1/4 note at about 120bps using the kick drum sound:

```
MidiOut  PIN 15             ' MIDI Output
MidiBaud CON $8000 + 12     ' 31.25 kBaud -- open
Channel  CON 8              ' MIDI Channel
NN       CON $90 | Channel  ' note on
NX       CON $80 | Channel  ' note off

AGAIN:
SEROUT MidiOut, MidiBaud, [NN,$24,$7f] 'NoteOn  kick note# 36 at full velocity
PAUSE 300
SEROUT MidiOut, MidiBaud, [NX,$24,$7f] 'NoteOff kick note# 36 at full velocity
PAUSE 300
GOTO AGAIN
```


Figure 11: Thereclock Finished and mounted.

I decided that it would be a good idea to mount the Thereclock in a central location to make it as easy as possible to connect the instruments. Also, since the MIDI output of the Thereclock would be sent to the drum machine, it seemed simplest to mount the Thereping directly onto the electronic drum set that held the MIDI module. This would allow me to play the drums to add “fills” over the straight MIDI beat coming from the Thereclock box. I had a 12” round wooden disk that would make a good base for the Basic Stamp Homework board as well as the RJ-45 jacks in the wall plate. The circular shape would make it fit in with the cymbals and allow attachment to a cymbal holder by simply drilling a single hole in the edge of the board. I built the prototype and tested it with the Therepings. It was successful in syncing up all the Thereping units!


Figure 12: The Thereping rehearsal

However, it became apparent that it would be necessary to find a way to STOP and START the unit to designate the beginning and ending of “songs”. I took some PC case back planes and bent them to fit on the side of the unit facing the drummer, and then installed and labeled

four pushbuttons to use for controls. Now I had the ability to START and STOP the song, and I also added the traditional “whistle” 8 beat “intro” and “outro” that would traditionally be used by a drum major to start and stop a marching band. Since the button's use is determined by software, this would also allow me to repurpose the functions in the future (i.e. Song select, tempo adjustment etc.). I also decided to remove the 1/4” Right and Left audio jacks and replace them with a single 1/8” female jack so the audio could be sent directly to either a small “computer speaker” type amplifier or using a 1/8” to RCA adapter, sent to the line input of a mixing console.

On the mixer, I panned the first three Therepings inputs to the left and the second set to the right, then added some echo and a touch of reverb. I didn't implement the audio out LPF circuit shown in the Basic Stamp manual because the consensus from people who had heard the prototype is that the “buzzy” sound was more appealing than the “duller” sound that came out after the LPF was applied. The finished Thereclock unit mounted nicely between the two cymbals on my Yamaha DTXpress drum kit (figure 11). The finished schematic for the Thereclock is shown in figure 10. I invited the The Robot Group members over and we had one single rehearsal at my house (Figure 12) and then we were “ready” for the debut performance before an expected crowd of well over 60,000 people! Yikes!

The Debut

On the morning of December 31st, 2005 members of The Robot Group arrived to load all the equipment onto a 20' trailer. After an interesting trip across town with a literal truck load of technical equipment, we arrived at the pavilion where we began to setup the PA system and the rest of the Thereping equipment. We immediately drew a crowd by playing for a short while (Figure 13, 15), and then we offered the Thereping units to people in the crowd so they could experiment with music. It took only a few moments to instruct someone on how to play the instrument. We had a crowd of people that watched the “jam session” for a good solid 3 hours! There were TV cameras from both local and national news organizations present. The most exciting part was the people participating. After just a few moments of instruction on what the parts of the instrument were, folks were able to pick up the instrument and start playing music together instantly! This part of the show was a major success, but we still had to tear down the entire system and reassemble it on the trailer for the parade.

Figure 13: Nic Graner playing Thereping on the Pavilion

Figure 14: Thereping Band ready to play in the parade!

Show time!

Once everything was hooked back up (and amazingly all the systems survived the relocation) we carefully drove the trrailer to the start of the parade route. We powered the entire show using a 2.5Kw gasoline generator on the bed of the pickup truck and put the 400 watt stereo PA system on the trailer with the musicians. Each of us wore a fanciful “digital” hat crafted by Denise Sciloi and others of The Robot Group (Figure 14). We played music for over an hour and had people clapping and dancing alongside the trailer as we went down Congress Avenue in Austin. First Night Austin officials estimated that over

100,000 people viewed the parade that night. It was an amazing experience! I think we fulfilled our mission to create a fun and interesting spectacle.

Resources:

Vern Graner's "Thereping" web site:

http://www.thereping.com

The Robot Group

http://www.robotgroup.net

First Night Austin

http://www.FirstNightAustin.org

Parallax Inc.

http://www.Parallax.com

Yamaha DTXpress User Group

http://www.DTXpressions.com

The Future

Though I consider this project a success as it stands, I also feel there is plenty of room for improvement for the equipment. For example, we've barely scratched the surface of the capabilities of the Thereclock to produce MIDI. With a bit more programming, it should be possible to create a number of different types of beats or even songs for playback.

The extra lines in the Ethernet cable could be used for sending song selection information to the Therepings or for allowing the Thereping to send a "stereo" signal to the Thereclock thereby enabling the player to select to which channel their signal is sent (i.e. Clean/distorted/echo or not etc.).

The patterns that can be chosen on the Therepings could also be expanded to include swing beats or triplets for example. The Drone setting could be altered to include MIDI output sent back via the cat-5 cable or to produce chords or blues-style "walking" bass lines. In short, this instrument has barely been touched in capability! If you do decide to build one I would love to hear how your project proceeds. My web site is supplied in the Resources area so you can find all the complete source code, schematics and associated material for helping build your own Thereping & Thereclock. If you have any questions feel free to contact me at vern@thereping.com.

Figure 15: The author and his son playing at First Night Austin, December 31 2005

Acknowledgments

I would like to thank the following people who were critical in making the Thereping a reality:

Rick Abbott

Paul Atkinson

Derek Bridges

Don Colbath

Bob Comer

Kym Graner

Nic Graner

Walt Graner

PY Hung

Eric Lundquist

Tom Morin

Gray Mack

Denise & Mike Scioli

Sharon Sudduth