Things to know about the Tellingent

#1
When running and touching a switch or sometimes the ‘Ball’ itself, will sometimes cause the program to stop. This is because the Electric Field is so powerful that your body will absorb and reradiate the Electric Field it is receiving. The effect this can have on the Arduino MPU is to cause a “glitch” where the program gets lost or misses a program step. Think of it like a mini EMP (Electro Magnetic Pulse). Without buying a military version of the Arduino (EMP harden electronics), or moving the Arduino into a separate housing, there is not much that can be done about this. Maybe in a ‘next life’ there will a way around this. After over a hundred hours of testing, running, and playing, this was not found to be significant enough issue to cancel this project. Because this is an OPEN SOURCE project maybe there will be a clever user that can help eliminate this “feature” ☺

#2
The internal Timers are ever so slightly off from being perfect. The timing features are based on the Arduino clock speed verse a “Real Time Clock” circuit. So a minute might actually time out as 61 or 62 seconds. Possibly could also go the other way to 59 or 58 seconds. Additionally when switching frequencies in PROTOCOL or SWEEP Mode, the software time to switch from one frequency to another is not part of the ‘timer count-down’ sequence. This can have an effect of making the program run longer than the requested time. For example a Protocol might be set to run for 2 hours and it actually runs for 2 hours and a few minutes. The active frequency will be close to 2 hours but the frequency to frequency switching time is not included in the ‘count-down’ time.

#3
Channel 2 or “F2” is used for gating. It also uses a low resolution 256 bit timer within the Arduino. This low resolution timer has a range of 31.5 hertz to 8300 hertz in this software version. However “F2” can actually perform gating as low as .01 hertz. A special software routine was written to produce very accurate gating frequencies from 0 .01 to 31.5 hertz. So when a frequency like Schuman Resonances of 7.83 is dialed in, it will be extremely accurate with respect to 7.83 hertz.

#4
Channel 1 or “F1” is used as the primary run frequency. It is digital based and uses a high resolution 65536 bit timer within the Arduino. So in theory 1 bit can represent 1 cycle or hertz. The Frequency range of the Plasma has been limited to 50000 hertz not because it couldn’t go higher, but because the hardware of the Plasma Ball cannot respond properly to frequencies above 50K. When running from the internal electronics, the software monitors everything and will impose limits to protect everything. When running from an external source like a Spooky2, **DO NOT RUN FREQUENCIES HIGHER THAN 50K**. Use the “Frequency Limit” function in the Spooky software to insure you stay below 50K.

#5
Channel 1 or “F1” Duty Cycle has some limits. This was done to insure the Plasma Ball driving MOSFET (transistor type device) does not get TOO HOT. The following table applies;

Less than 5000 hertz has a 20% maximum
Less than 4000 hertz has a 15% maximum
Less than 3000 hertz has a 10% maximum
Less than 1000 hertz has a 5% maximum
Less than 100 hertz has a 3% maximum
Less than 50 hertz has a 0% maximum

The software will over-ride setting outside the above table range. As can be seen by the table, Channel 1 cannot run the Plasma Ball lower than 50 hertz. It is suggested that gating be used to run very low frequencies. (See Operational Video 1 for more information).

NOTE: When running from a Spooky2 or other function generator, use the same table limits.