

```

// Ultracap Charger
// Control Unit to charge a Ultracapcitor bank of 16 Volt with a constant current
charger
// Using an Arduino NANO
// History v1.0 First working & tested version (on work bench)
// History v1.1 Changed into exact same voltage - not with the delta anymore (and
add LED 13 light up when voltage is achieved) - and no delay in loop anymore
// History v1.2 This version changes at 12.2 Volt versus 12.5 V - this 0,3 Volt
accounts for 102 value from 1024 of the 10 bits ADC

/*
=====
=====
This Control Unit to charge a Ultracapcitor bank of 6 Ultracaps 2,7V with a
constant current charger
Start up is a 4 x relay bank Arduino style- Low is active / High is power off
position
After the UC Bank is charged up to the same (~12,2 Volt) voltage (MINUS 0,3
Volt) as the "normal" PSU
The voltages are measured through a voltage divider of 1/3rd of PSU Voltage (Rs
10k Rp 5k and 1uF filter)
the relais will switch from the 10A lcharger to the standard PSU and
the relais will also switch OFF the charger mains power
and also will switch the control input to LOW by itself
The Charger GND will be depowered by a relay
Only a full power reset will bring back the 10A CC charger to the bank and
power Arduino
This is a "one tiome action" code !!
Digital Output is driving a 4x relay bank

So far no safety measures for all kind of situations which are unlikely to
happen any way

=====
=====

*/
// ***** CODE STARTS HERE *****

// set pin numbers and all variables
const byte UltraCap_pin = A0; // Input Pin A0 for analog reading of
1/3rd of the voltage of the Ultracap Bank
const byte DDDAC_PSU_pin = A1; // Input Pin A1 for analog reading of
1/3rd of the voltage of the DDDAC PSU
const byte Relay_ON_OFF_1 = 2; // Output Pin D2 to control the Relay
module Nr 1 - switch mains of 10A charger

```

```

const byte Relay_ON_OFF_2 = 3;          // Output Pin D3 to control the Relay
module Nr 2 - switch DDDAC PSU to Ultracap (and hence 10A charger PLUS disconnect)
const byte Relay_ON_OFF_3 = 4;          // Output Pin D4 to control the Relay
module Nr 3 - switch GND - 10A Charger
const byte Relay_ON_OFF_4 = 5;          // Output Pin D5 to control the Relay
module Nr 4 - Connect GND DAC and GND Arduino
const int delta            = 21;        // Output volt delta when the charger is
decoupled. 21 is the A/D equivalent of 300mV ...

// set Global variables
.....
byte Status = 0      ;                  // This is a status flag used to "disable
the loop" (when Status = 1) after the switch sequence has been started (which is
once only)
int  Read_UltraCap  ;                  // help variable for remembering IR code -
start with "reset code"
int  Read_DDDAC_PSU ;                  // help variable - "0" means the unit is
not active

//
*****setup*****
*****

void setup()
{
  pinMode(UltraCap_pin,      INPUT);    // Input Pin for reading Peak detector
signal from Revox (via extra DIN 5 bus)
  pinMode(DDDAC_PSU_pin,     INPUT);    // Input Pin for reading Peak detector
signal from Revox (via extra DIN 5 bus)
  pinMode(Relay_ON_OFF_1,    OUTPUT);   // set PinMode
  pinMode(Relay_ON_OFF_2,    OUTPUT);   // set PinMode
  pinMode(Relay_ON_OFF_3,    OUTPUT);   // set PinMode
  pinMode(Relay_ON_OFF_4,    OUTPUT);   // set PinMode
  pinMode(13,                OUTPUT);   // On board LED for control mode
  // First actions:
  digitalWrite(Relay_ON_OFF_1, HIGH);   // Start with relay in the "OFF" mode
(=HIGH)
  digitalWrite(Relay_ON_OFF_2, HIGH);
  digitalWrite(Relay_ON_OFF_3,  LOW);   // At power down the MINUS Pole of the
charger is NOT connected to avoid the UC to discharge back - This LOW command is
now activating the relay and connecting it and the charging starts
  digitalWrite(Relay_ON_OFF_4, HIGH);
  digitalWrite(13,          LOW);        // On board LED = OFF
  delay (1000);                        // Give the DDDAC PSU time to settle
before we start any read measurements
}

```

```

//*****loop*****
*****
//*****loop*****
*****
//*****loop*****
*****

void loop()
{
  if (Status == 0) // Check if status is NOT 1
yet, so we keep reading the UltraCap
  {
    Read_DDDAC_PSU = analogRead(DDDAC_PSU_pin); // Read the value of the
DDDAC PSU
    if (Read_DDDAC_PSU > 680) // Only go on when the
voltage of the PSU is above 10 Volt) = 680 10 bit A/D - this value is arbitrary
and only like a an enabler
    {
      Read_UltraCap = analogRead(UltraCap_pin) ; // Read the value - the
range from 0-1024 is equal to 0-15 Volt as of the 1/ 3 voltage divider at the
input (10 - 4k7)
      if (Read_UltraCap > Read_DDDAC_PSU - delta) // Check if the two voltages
are close enough to switch to connect UltraCap to to DDDAC PSU modus - taking
into account that the PSU will be 0,3 volt higher
      {
        digitalWrite(13, HIGH); // On board LED = ON
        digitalWrite(Relay_ON_OFF_1, LOW); // Switch OFF the mains
power of the 10A Charger
        delay (10);
        digitalWrite(Relay_ON_OFF_2, LOW); // Switch the PLUS of the
ultracap to the DDDAC PSU
        delay (10);
        digitalWrite(Relay_ON_OFF_3, HIGH); // Now also Disconnect the
MINUS POLE from the Charger from the DAC Ground
        delay (10);
        digitalWrite(Relay_ON_OFF_4, LOW); // Disconnect the Arduino
Ground from the DAC Ground
        Status = 1; // With this status we
indicate that the Ultracap is FULL and is now connected to the normal PSU
      }
    }
  }
}

// All Code ends here

```