In this issue

CQ-DATV dotMOBI

Like us on Facebook

Production Team

Ian Pawson G8IQU Trevor Brown G8CJS
Terry Mowles VK5TM

Contributing Authors

Trevor Brown G8CJS John Hudson G3RFL
Ken Konechy W6HHC Klaus, DL4KC
John Owen

Editorial

So it's May and the weather in the UK has not quite made up its mind. If it's warm and out with the portable kit or still cold and blustery, keep the shack door shut and the soldering iron on.

John G3RFL is obviously in the keep the soldering iron on camp, as he has also been working on a solution to the dreaded blue screen on poor signals that stops you seeing pictures and improving them, particularly when using a TFT screen.

John's fix for the problem is slightly more sophisticated than Dave's solution (CQ-DATV 46), in that it uses more parts and surprise, surprise, has a PIC generating pseudo sync pulses to fool the monitor out of blue screen mode and the when the picture is improved to the point when the blue screen mute is overcome, it automatically switches to incoming syncs. Johns version has several adjustments as the tipping point of this blue mute is different on all monitors. Also as you would expect from John, there is a single sided PCB layout to aid construction.

We also have a new project in Micro Corner and this time it's an VGA to AV Converter. It converts a VGA signal from a computer to an AV and S-Video signal for use by a TV, monitor or a PC video capture device. This is a commercial device but has been modified by John Owen to interface to an Arduino Nano, so two projects for you to get your teeth into in this issue.

Ken has delivered his latest DATV express update and reports on three outstanding problems :-

Windows-for-Italian OS does not select capture-camera properly because of use of a "Italian-friendly" setting name. Using vMix should be a work-around.

Windows-for-Japanese OS does not select capture-camera properly because of use of a "Japanese-friendly" setting name. Using vMix should be a work-around.

The MER readings using MiniTiouner receiver for DVB-S2 transmissions appear to read "lower" in value than when using the DVB-S protocol, due to an "unknown reason".

In the firm belief that the weather will improve at least in the UK can we remind everyone that the IARU region 1 ATV contest will be start at 12:00 UTC on the 10th of June and runs until 18:00 UTC on the following day (June 11). So enough time to schedule and make that important ATV contact or better still go portable.

Most operators do not really consider this event as a contest, but as an 'activity weekend', many ATV operators in several countries, will be active over this weekend, with a lot of portable operation.

Please use this opportunity to push activity on the higher microwave bands. We need to keep these bands active, do not complain about lack of activity; you are the one that must make activity happen!

Last year, stations from Sweden joined in, some only made one contact but put ATV on the map for their country. Spain promised to join this year, so please point your aerials in their direction.

Please see the 2016 news letter: *https://www.iaru-r1.org/images/VHF/newsletters/Newsletter_70.pdf*

The Contest rules are simple, and an Excel log sheet needs to go to your local contest manager within 2 weeks. If you do not have a local ATV contest manager, then the logs can be sent direct, full details on the site.

https://www.iaru-r1.org/images/VHF/atv/IARU_ATV_contest_rules_version_2015.pdf (English Version)

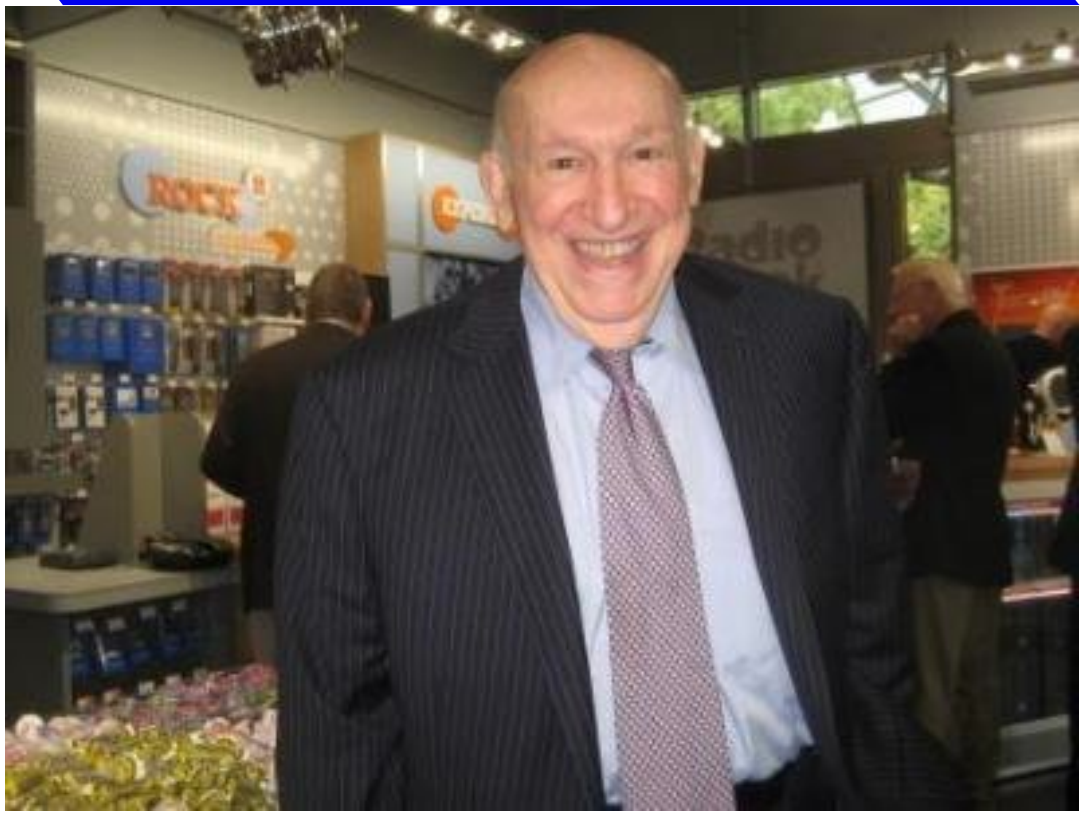
https://www.iaru-r1.org/images/VHF/atv/ATV_contest_log_-_ATV_yourcall_YYYYMMDD.xls (English version)

Please be aware the four digits should only be exchanged in vision, do not show your code on the internet.

Also strongly recommended that ATV repeaters should be switched off during this activity weekend to make it possible to use the whole band (and to prevent confusion in case of retransmitted four digit codes).

CQ-DATV Production team

DATV News



Sadly on the 2nd of April Bernie Appel passed away.

Mr. Appel is one of the distinguished members of the Consumer Electronics Hall of Fame, which includes Alexander Graham Bell, Ray Dolby, Thomas Edison, but perhaps better known as the motivating force behind Radio Shack. Although Bernie did not found Radio Shack, he did join the Boston-based catalogue business company in 1959 as a buyer. He rose to the chairman and was the force behind the establishment of over 7,000 tech stores across the USA.

His work earned him the moniker “Mr. RadioShack,” as well as a twice Distinguished Achievement Award in 1994, and entrée into the Consumer Technology Hall of Fame in 2002 and has frequently said it was his most memorable *industry moment*.

Appel was welcomed into the Consumer Technology Hall of Fame in its third year by CTA head Gary Shapiro in 2002

Appel left RadioShack and RadioShack would go on to face mounting hardships as it searched for a contemporary identity and compelling role in the digital age, and last month filed for Chapter 11 *bankruptcy protection* for the second time in as many years.

WD0FCH DVB-T repeater

A 70cm in-band DVB-T repeater, WD0FCH/R is now on-the-air in St Louis, Missouri, EM48.

WD0FCH DVB-T repeater, St. Louis, MO

• Band	441.00 MHz input frequency
• Output Freq	423.00 MHz output frequency
• Filters	N0YE Inter-digital RX/TX Bp
• Video encoding	H.264/AVC
• Channel BW	6 MHz.
• FEC	2/3
• Guard spacing	1/16
• Modulation	QPSK
• FFT	8K mode
• Modulation Data Rate	5.85 Mbps
• Video aspect ratio	16:9
• Audio Encoding	96 Kbps MPEG2
• Video PID	1601(base 10) - 0x641
• Audio PID	1602(base 10) - 0x642
• Transmitter	HIDes HV-100EH w/remote web management
• Receiver	HIDes HV-110Mel, K0PFX

Ken W6HHC believes this is now USA DATV-repeater number 15. See CQ-DATV issue 46/April for an earlier list of DATV Repeaters in USA.

VGA to AV Converter Modification & Switcher

By John Owen (*http://www.vwlowen.co.uk*)
Reproduced here by kind permission.

The VGA to AV Converter shown above left is available from dozens of retailers. It converts a VGA signal from a computer to an AV and S-Video signal for use by a TV, monitor or a PC video capture device. It also provides an unaltered feed-through for the VGA signal

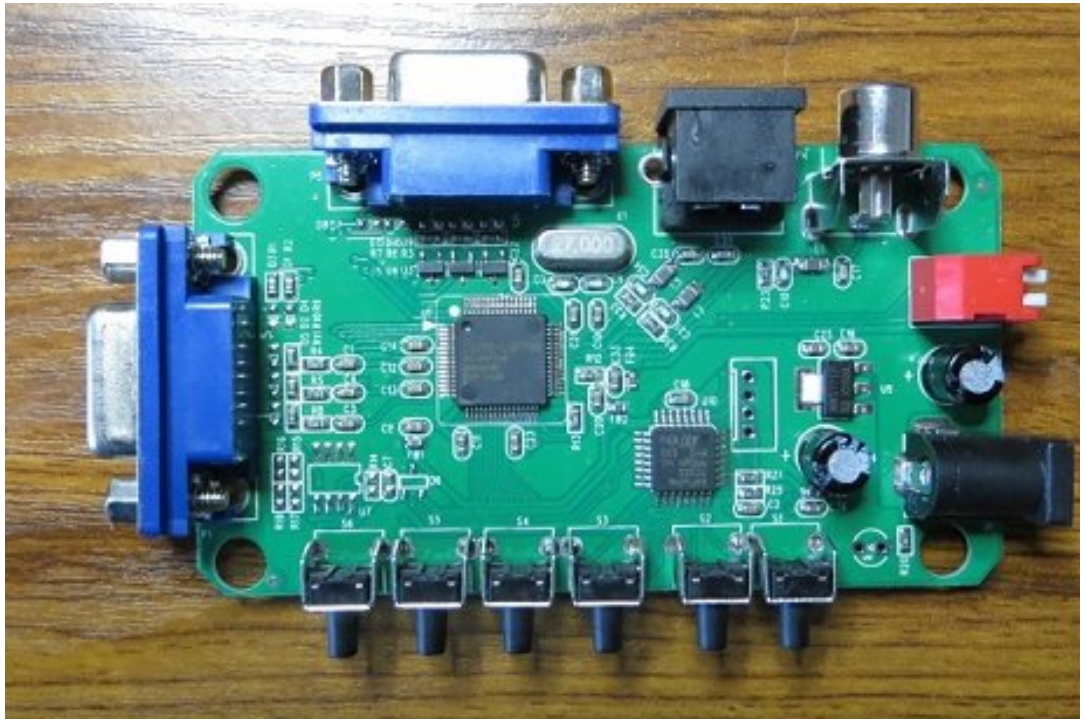
A common use is with a CCTV Digital Video Recorder (DVR). Many of these units only provide a VGA output (and, optionally, HDMI) to a monitor but an additional AV output can be useful, for example, to feed a TV (with an AV input) or to feed a video capture device in a PC - which nearly always involves AV to USB capture hardware or a PCI(e) card with AV inputs (Such as the *Osprey range* of analogue video capture cards).

The VGA to AV Video Converter unit has a useful button on its front panel (called ZOOM) successive presses of which cycles between passing the full VGA screen to the AV output or each of the four "corners".

CCTV DVRs frequently display video from up to four cameras on-screen at the same time so the ZOOM button can be used to select either the full screen or one of the four camera images, as shown below.

The VGA image always passes through the converter unchanged and displays whatever output format is selected on the DVR but I wanted to automate the ZOOM button-pressing process so that the AV image could be cycled either on a timer or even switched remotely via a web page. For completeness, I also wanted to retain the ZOOM button.

Inside the VGA to AV Converter

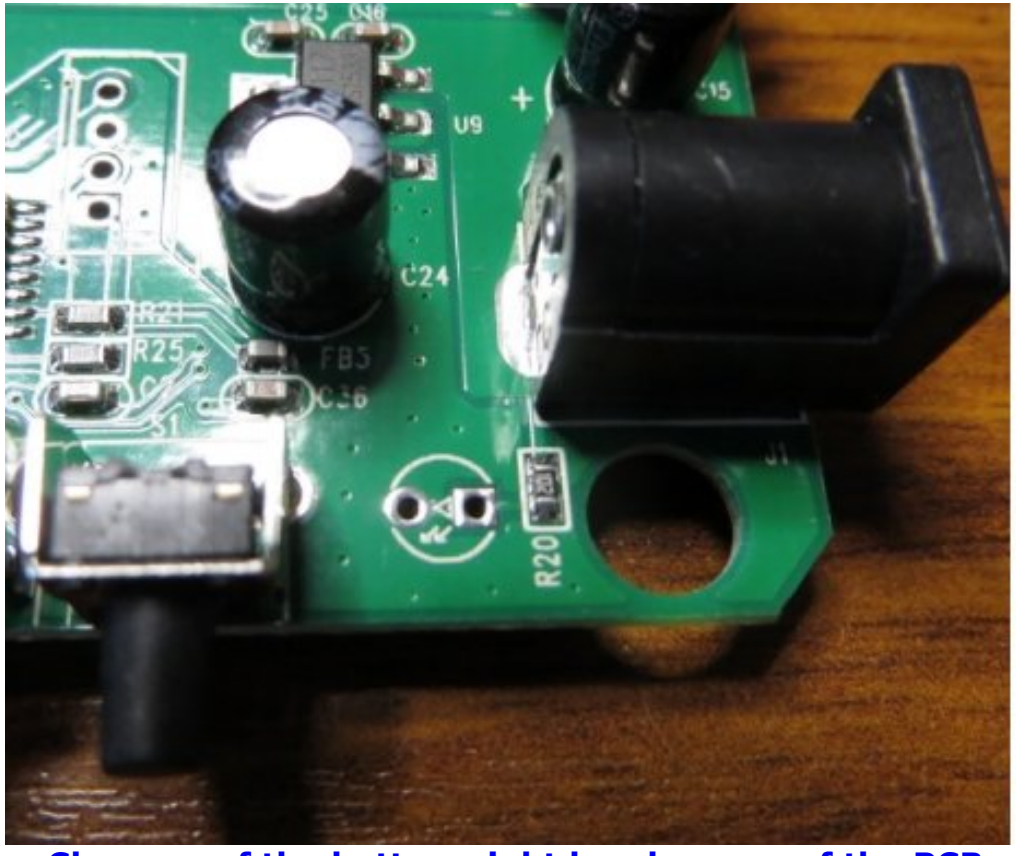


The converter's naked PCB

To gain access to the converter's PCB, it's necessary to remove four screws on the underside. It's also necessary to carefully peel off the label surrounding the button panel to allow the buttons to ease clear of the case.

The idea is to pick up the connections on the ZOOM button and wire them to a small socket for connection to an external switching circuit/microprocessor or relay.

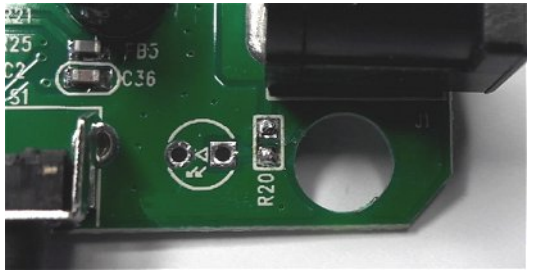
The ZOOM button is the one on the bottom right corner in this photo. The ZOOM button (in fact, all of the buttons) has one side permanently connected to ground and pushing the button obviously grounds the other connection. As you'll see, there isn't much room anywhere around the edges for an additional socket.

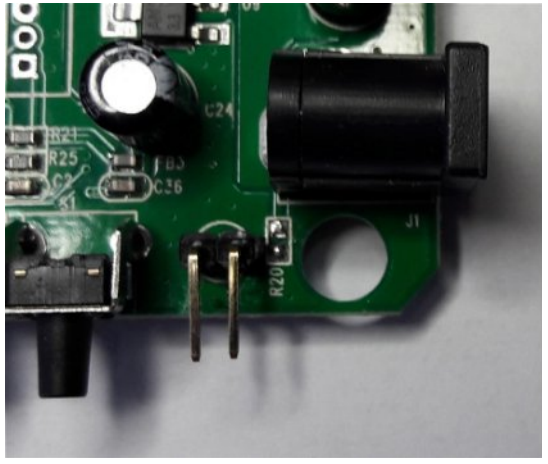


Close-up of the bottom right hand corner of the PCB

This close-up of the bottom right hand corner of the PCB shows spare pads where an LED could be. R20 is a 1k resistor connected between one side of the LED (on the square solder pad) and the main power input socket. The other pad for the LED connects to ground. This is clearly an unused POWER ON LED and the pads are ideal for our purpose.

As the photo below left shows, the square pad is isolated from everything by simply removing R20 with the soldering iron.





Th photo above right shows a 2.5mm right-angled header soldered in place of the LED, while the photo below shows a 1k resistor soldered between the two non-grounded pins - one on the pin-header and one on the ZOOM button.

A resistor is used to prevent potential damage to the output of an external microprocessor in case the ZOOM button is pressed while the microprocessor output is HIGH.

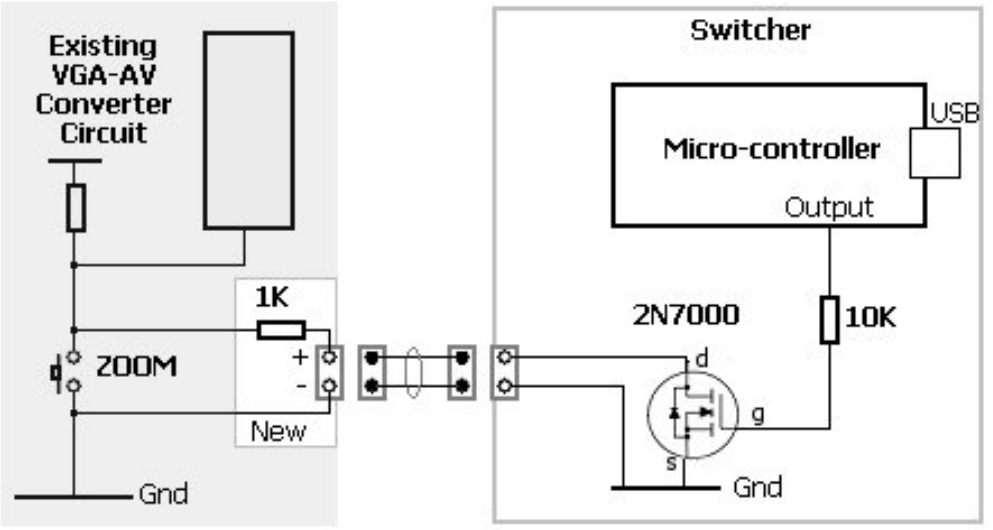
Note that the electronics within the converter operates at 3.2 volts.

The absolute maximum voltage that should be applied to the ZOOM input (according to the data sheet for the main HY57V161610 processor) is 4.6 volts so it makes sense to design our external switch to operate on 3.2 volts or use a MOSFET (or transistor) to do the switching.

The photo below shows the converter in use with the PCB back in its case after filing a new opening for the right-angled pin header and replacing the stick-on label.

Note that I marked the polarity of the two header pins as it's obviously important to avoid connecting the external microprocessor output to the ground pin.

The Arduino Switching Hardware



The basic switching circuit is shown above. I used an Arduino Nano (with an ATMEL mega328p chip) rather than a discrete ATmega328 because, as the project also needs a serial USB connection to a PC running custom internet server software, it's more cost-effective and convenient to use a standard Nano module which includes the USB hardware.

The internet server software running on the PC serves a continuously-updated image (captured from the VGA-to-AV converter) to a web page and listens for "switching requests" from the webpage.

The switching requests take the form of HTTP POST requests, when a visitor presses a SUBMIT button, and consist of a 4-digit code key.

If the requested code matches a user-configured code key in the server software, the server sends the key via a serial USB port to the Arduino switching hardware.

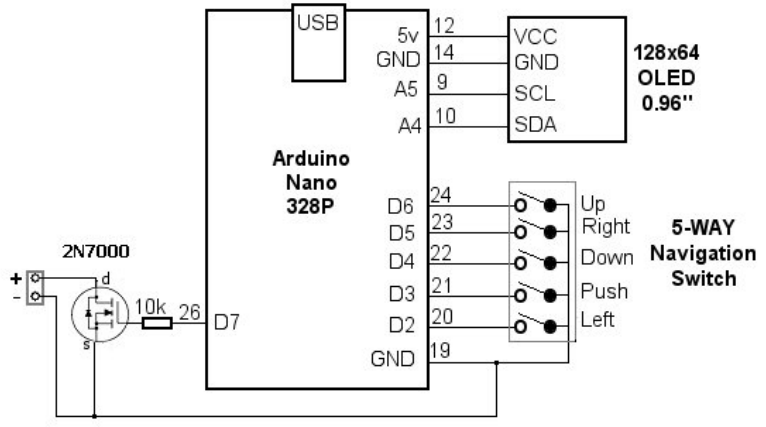
It should be noted that this isn't a secure "transaction" because the necessary code key is in plain view in the web page's source code. However, the server software does provide for standard Username & Password HTML authentication for the web page should this be required.

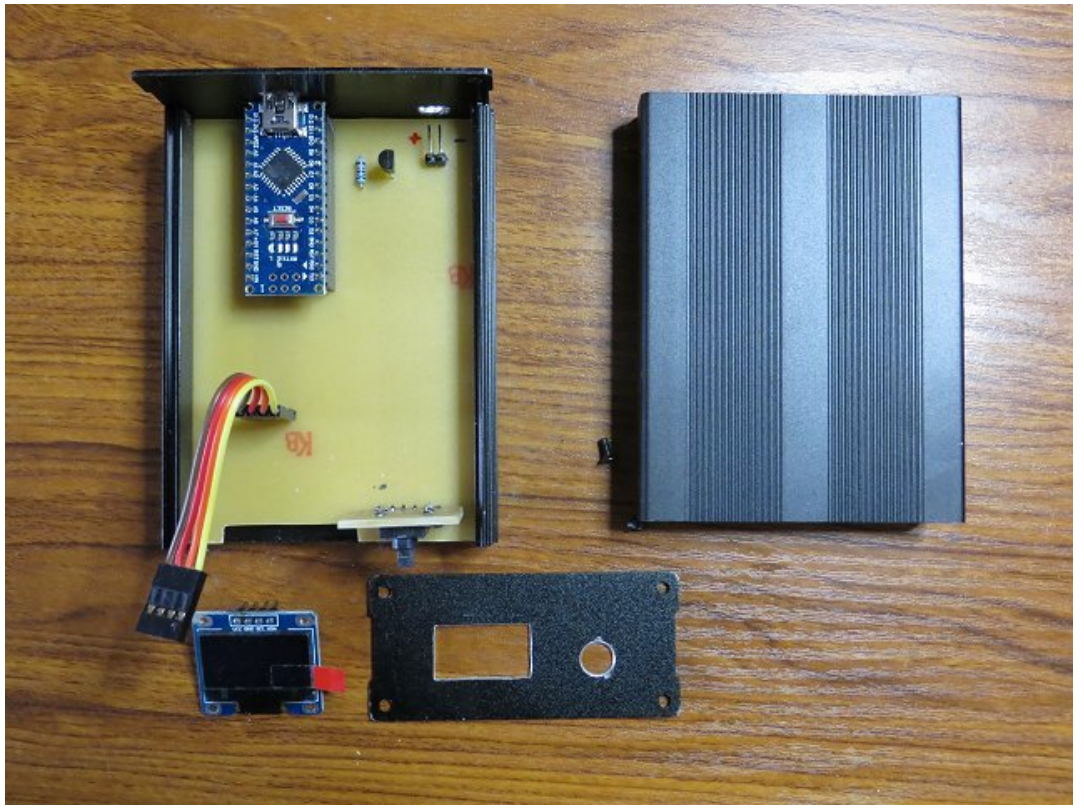
If the key matches a user-settable one configured in the Arduino software, an Arduino OUTPUT sends a short 5v pulse to the 2N7000 Mosfet. This simulates pressing the ZOOM button on the VGA to AV Converter which, in turn, cycles to the next quadrant's image as explained above.

As the Arduino software is so simple, it seemed sensible to add a few extra features. So I added a simple push button to replicate pressing the VGA-to-AV converter's ZOOM button and an adjustable timer to cycle through the quadrant images automatically at a pre-set interval.

I also wanted a way to change the code key in the Arduino software so that more than one unit could be used at the same time without having to program "bespoke" code keys into each unit at construction time.

Both the adjustable timer and the ability to manually set the code key meant that some sort of display was needed so the project had suddenly expanded into the circuit below.





The PCB artwork in PDF format can be downloaded from here *http://www.vwlowen.co.uk/arduino/vga-switcher//vga-switcher.pdf* and the PC server software (Windows 7, 10) here *http://www.vwlowen.co.uk/mjpeg/mjpeg-cam.htm*

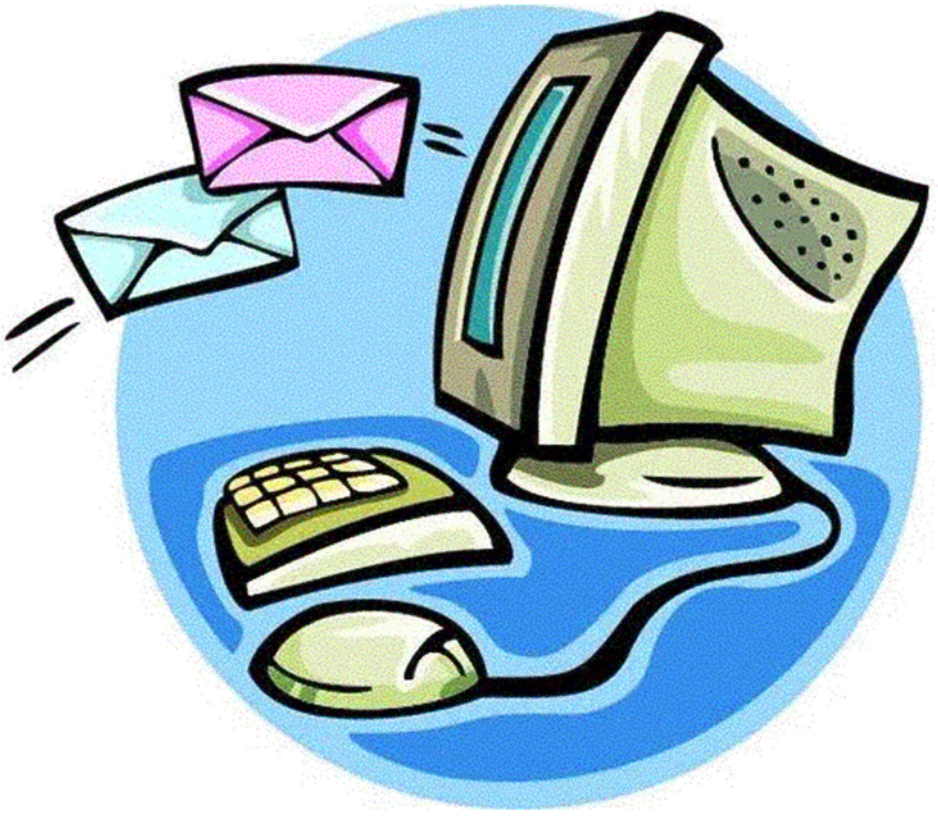
Arduino code

```
#include  <SPI.h> include  <Wire.h> include
<Adafruit_GFX.h>                    //
https://github.com/adafruit/Adafruit-GFX-Library
#include  <Adafruit_SSD1306.h>
//https://github.com/adafruit/Adafruit_SSD1306

#include  <EEPROM.h> #define OLED_RESET 9
// Our display doesn't use RESET so use a spare IO
Adafruit_SSD1306 oled(OLED_RESET);
```

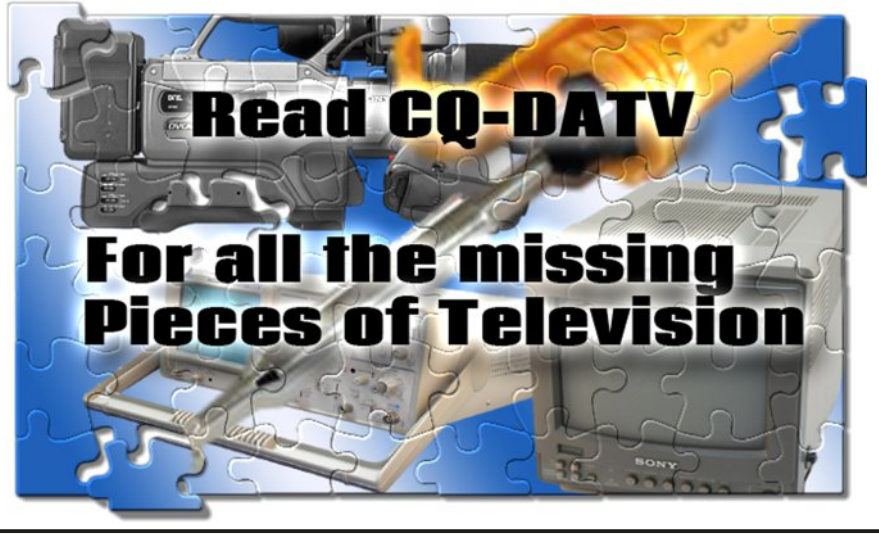
```
#define pulseOut  7

#define Left  2
#define Push  3
#define Down  4
#define Right 5
#define Up    6

String key;              // This is the value sent by the VGA-
Switcher software when it requests
int keyNumber;           // a picture 'ZOOM'.  We convert it to
an interger to save it in EEPROM.


byte mode = 35;          // Mode indicates which value will
be changed by the  <  and  >  buttons.
                         // 0 = key value will be changed. 35 =
timer value. Why not 0 and 1?
                         // 35 is simply a convenience as it is the
'Y' value on the OLED display.

int tim = 0;             // Switching timer value saved in
EEPROM.
unsigned long timerPreset; // 'tim' value (seconds) converted
to milliseconds.
unsigned long timer;     // Current value of the running
timer in milliseconds.

unsigned long holdTimer;         // Time in milliseconds a
button is held. Increment rate increases...
unsigned long holdPreset = 5000;   // .. after holdPreset time
(msec).

void setup() {

  keyNumber = constrain(eepromReadInt(0), 0, 9999);   //
Get key and timer values from EPROM.
  tim = constrain(eepromReadInt(2), 0, 600);
  timerPreset = tim * 1000;

  key = String(keyNumber);       // Convert key value to
String for comparison with received string.

  pinMode(pulseOut, OUTPUT);     // Set inputs and poutput
pin modes.

  pinMode(Push, INPUT_PULLUP);
  pinMode(Up, INPUT_PULLUP);
  pinMode(Down, INPUT_PULLUP);
  pinMode(Left, INPUT_PULLUP);
  pinMode(Right, INPUT_PULLUP);

  Serial.begin(9600);

  oled.begin(SSD1306_SWITCHCAPVCC, 0x3c);    // Start
the OLED

  oled.setTextSize(2);
  oled.setTextColor(WHITE);
  updateDisplay();                    // Display values
obtained from EEPROM.

  timer = millis();                   // Initialize the switching
timer.
}

void loop() {
 if (digitalRead(Push) == LOW) {          // If Push button is
pressed, send a pulse.
    sendPulse();
    while(digitalRead(Push) == LOW);
 }

 if (Serial.available()  >  0) {                    // If a
received string matches our key,
   if (Serial.readString() == key) sendPulse();          //
```

```
send a pulse.
  }

  if ((tim  >  0)  &&  ((millis() - timer)   >  timerPreset)) {
// If timer times out, send a pulse.
    sendPulse();
    timer = millis();
  }

  if ((digitalRead(Down) == LOW) || (digitalRead(Up) ==
LOW)) {   // Move cursor on OLED ro row 0
    mode == 0 ? mode = 35 : mode = 0;
// or row 35.
    updateDisplay();                                    // Update
the OLED display.
    while((digitalRead(Down) == LOW) || (digitalRead(Up) ==
LOW));
    delay(150);
  }

  if (digitalRead(Right) == LOW) {                  // If right
button is pressed, increase..
    holdTimer = millis();                          // the key value
if mode is 0 ir increase..
    while(digitalRead(Right) == LOW) {               // the
preset timer value if mode is 35.
      if (mode == 0) {
        keyNumber++;
        if (keyNumber  >  9999) keyNumber = 0;
        key = String(keyNumber);
      }

      if (mode == 35) {
        tim += 10;
        if (tim  >  600) tim = 0;
        timerPreset = tim * 1000;
      }
      updateDisplay();
      (millis() - holdTimer)  >  holdPreset ? delay(2) :
delay(200);
    }
    if (eepromReadInt(0) != keyNumber) eepromWriteInt(0,
keyNumber);    // Update values in EEPROM but
    if (eepromReadInt(2) != tim) eepromWriteInt(2, tim);
// only if they have changed.
  }

  if (digitalRead(Left) == LOW) {
    holdTimer = millis();
    while(digitalRead(Left) == LOW) {
      if (mode == 0) {
        keyNumber--;
        if (keyNumber  <  0) keyNumber = 9999;
        key = String(keyNumber);
      }

      if (mode == 35) {
        tim -= 10;
        if (tim  <  0) tim = 600;
        timerPreset = tim * 1000;
      }
      updateDisplay();
      (millis() - holdTimer)  >  holdPreset ? delay(2) :
delay(200);
    }
    if (eepromReadInt(0) != keyNumber) eepromWriteInt(0,
keyNumber);
    if (eepromReadInt(2) != tim) eepromWriteInt(2, tim);
  }



}

void updateDisplay() {
  oled.clearDisplay();
  oled.setCursor(0, mode);
```

```
  oled.print(">");
  oled.setCursor(15,0);
  oled.print("Key:");
  oled.setCursor(75, 0);
  oled.print(key);
  oled.setCursor(15, 35);
  oled.print("Time:");
  oled.setCursor(75, 35);
  oled.print(tim);
  oled.display();

}

void sendPulse() {
  digitalWrite(pulseOut, HIGH);
  delay(150);
  digitalWrite(pulseOut, LOW);
}

// http://www.elcojacobs.com/storing-settings-between-
restarts-in-eeprom-on-arduino/

int eepromReadInt(int address){
   int value = 0x0000;
   value = value | (EEPROM.read(address)  <<  8);
   value = value | EEPROM.read(address+1);
   return value;
}

void eepromWriteInt(int address, int value){
   EEPROM.write(address, (value  >>  8)  &  0xFF );
   EEPROM.write(address+1, value  &  0xFF);
}
```



Want to be notified when issues of CQ-DATV are published? Then join our *mailing list*.



DKARS MAGAZINE

In dit nummer:

- *De resultaten van de laatste radio amateurpanel vragen*
- *Nieuwe DKARS bestuurders*
- *Verslag 'Heelweg Microwave Meeting'*
- *Writing your own Logbook (6)*
- *And much more...*

Kingdom of the Netherlands

ISSN: 2452-1809

DKARS

Dutch Kingdom Amateur Radio Society

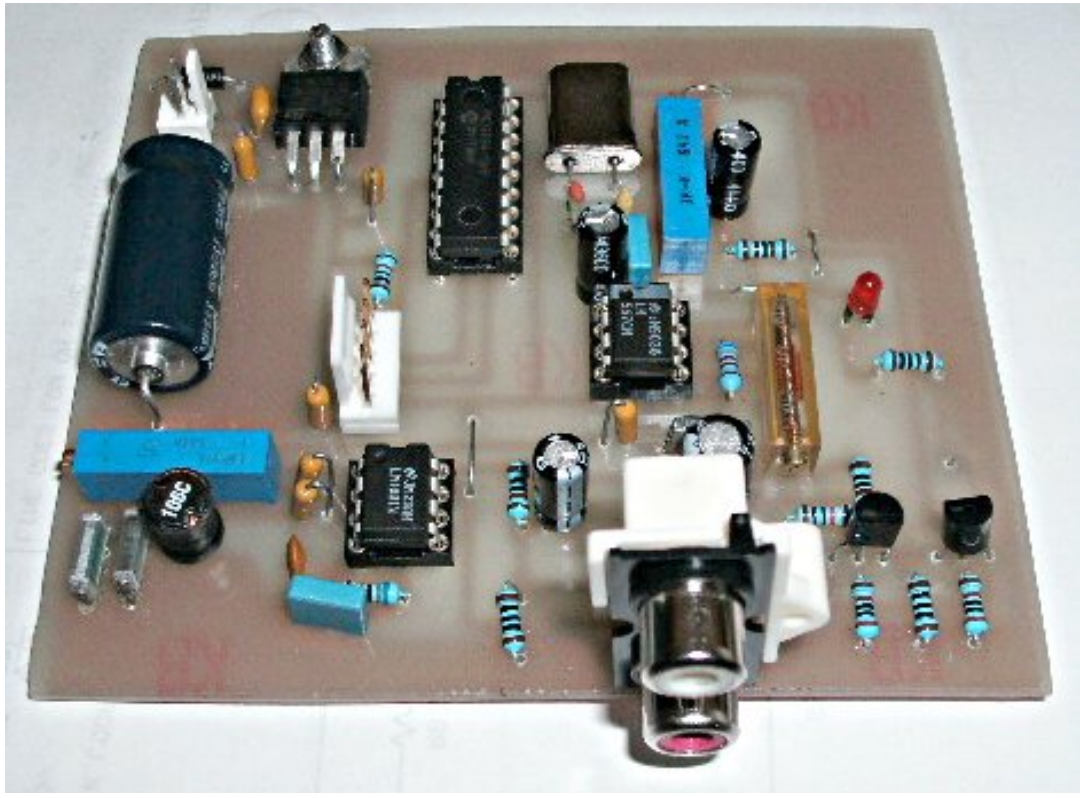
Prijs / Price € 0,00 / $ 0,00

Maart 2017 editie 31

Check out the DKARS website at:-
http://www.dkars.nl/

An automatic Blue Screen Eliminator

By John Hudson G3RFL



The Populated PCB of the Blue Screen Eliminator

It's not often, but it does happen, that two people can be working on the same problem at the same time and come up with two entirely different engineering solutions.

But that is the case. In CQ-DATV 46, Dave G3ZGZ came up with a Blue Screen Eliminator just as I was finishing my solution to the same problem. To make matters worse, yes we do talk to each other, often through our local ATV repeater GB3FY, but obviously not about blue screen elimination!

As Dave pointed out in CQ-DATV 46, the problem is the monitor usually looks for incoming line sync pulses and when these are found it will display the video signal. On very weak signals the sync pulses are normally poor and the monitor won't switch away from a blue screen.

This is exactly when it is necessary to peak antennas for the best signal and if you are operating portable it may pay to move the antenna slightly from its present position to take the benefit of a reflection etc.

Both blue screen eliminators work by tricking the monitor into thinking that there is a usable sync pulse being received and allow it to show an extremely noisy signal. Antennas and tuning can then be peaked for the best results.

Dave used the NE555 to generate sync pulses that would trick the internal detectors in the monitor to switch from blue screen mute. I did what I always do and programmed a PIC micro to do the same. NE555's do an excellent job, but are sometimes a little hard to find. Dave opted for manual switching to incoming sync when the monitor becomes happy with the detected video syncs.

I went down a different path and opted for an automatic switch. As you will see from the circuit diagram my version is a more complex solution - both circuits work well.

Either way both circuits generate local syncs that are not locked to the incoming video so at some point a switch from local to incoming video syncs is required,. This can be a manual or automatic decision, both circuits have their merits.

I used a LM1881 to strip the incoming video from the syncs and an LM567 phase locked loop to detect the presence of valid incoming syncs. I used a very similar video detector in the logic of GB3FY which is now in its 5th year of operation and so this is a well proven circuit configuration. It is proceeded by a simple filter that shapes the sync into a waveform more suited for the LM567 to lock to.

When the loop locks an LED illuminates to indicate syncs have been detected. The local pseudo pulses are stopped from leaving the PIC and negates the requirement for manual change over process.

The pseudo sync are added to the noisy video in the BC 547 long tailed pair to which I have added a pot so you can pre-set the tipping point for the switch. The unit works with +5 volts that is derived from a LM 7508 regulator.

The code used to programme the PIC as a local sync pulse generator is on the *CQ-DATV download* site in both Hex and ASM formats.

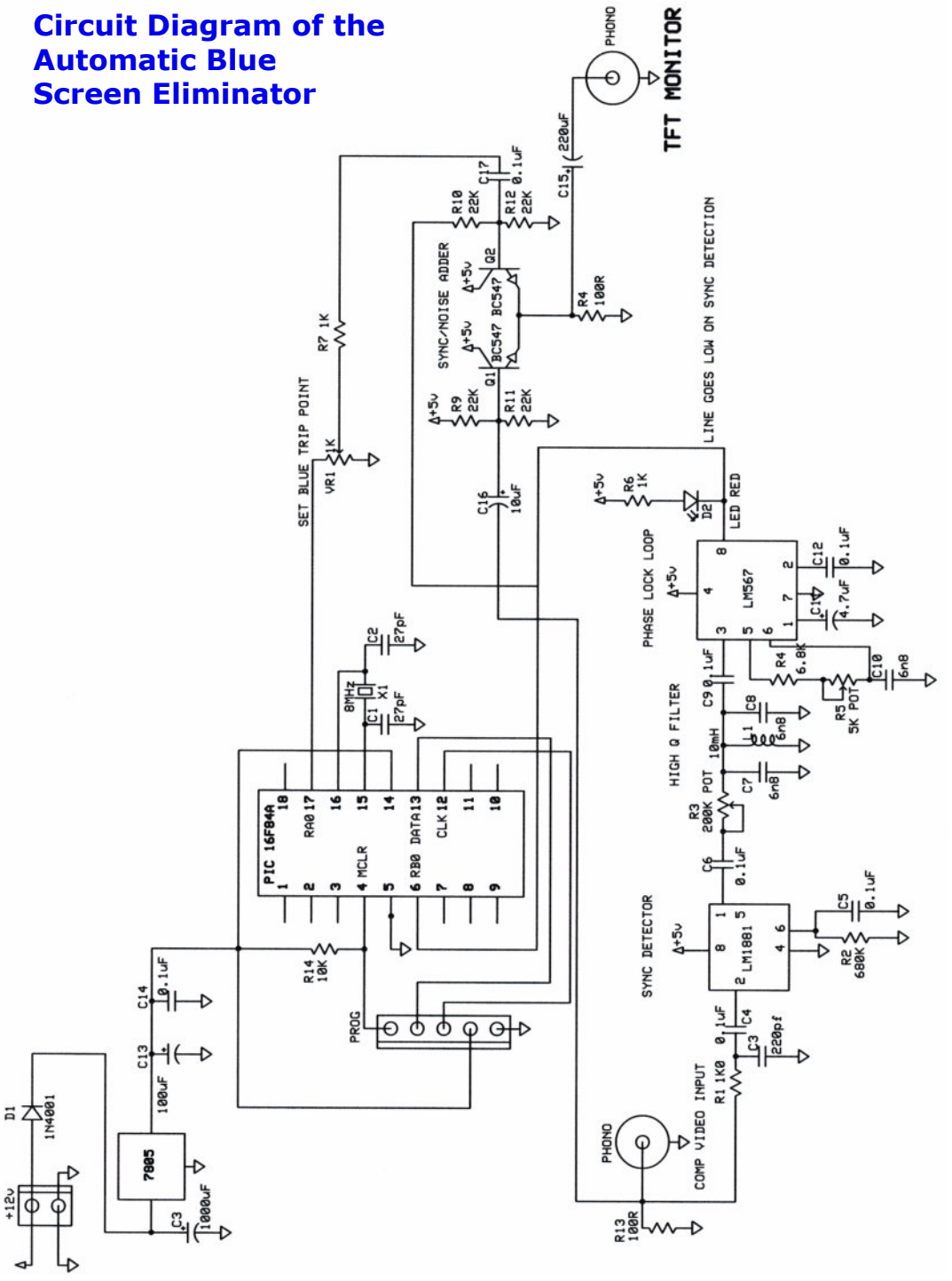
The PCB was a simple home etched single sided board and I have included the artwork for those of you that like etching your own PCB's. All the setup pots are multi turn, mainly because that is what in my junk box.

This unit can easily be constructed in a single evening and so far has been trouble free and very useful in the shack. It also has portable uses as it only requires a single rail power supply which can be 12 volts from a car battery as it has on-board regulation provided by a LM 7805. The components were easy to source. Perhaps the most difficult was L1 which came from CPC and has a part number PW00026.

The setting up was not too difficult as there are only three pots.

- *Set R3 to a mid value and place a video signal on the INPUT*
- *Adjust R5 until the LED D2 lights up. Keep going until it goes out, counting the number of turns at each point, then set in the middle.*
- *Remove the video input then increase VR1 until the video out locks a picture with a blank screen or use a scope to see it.*

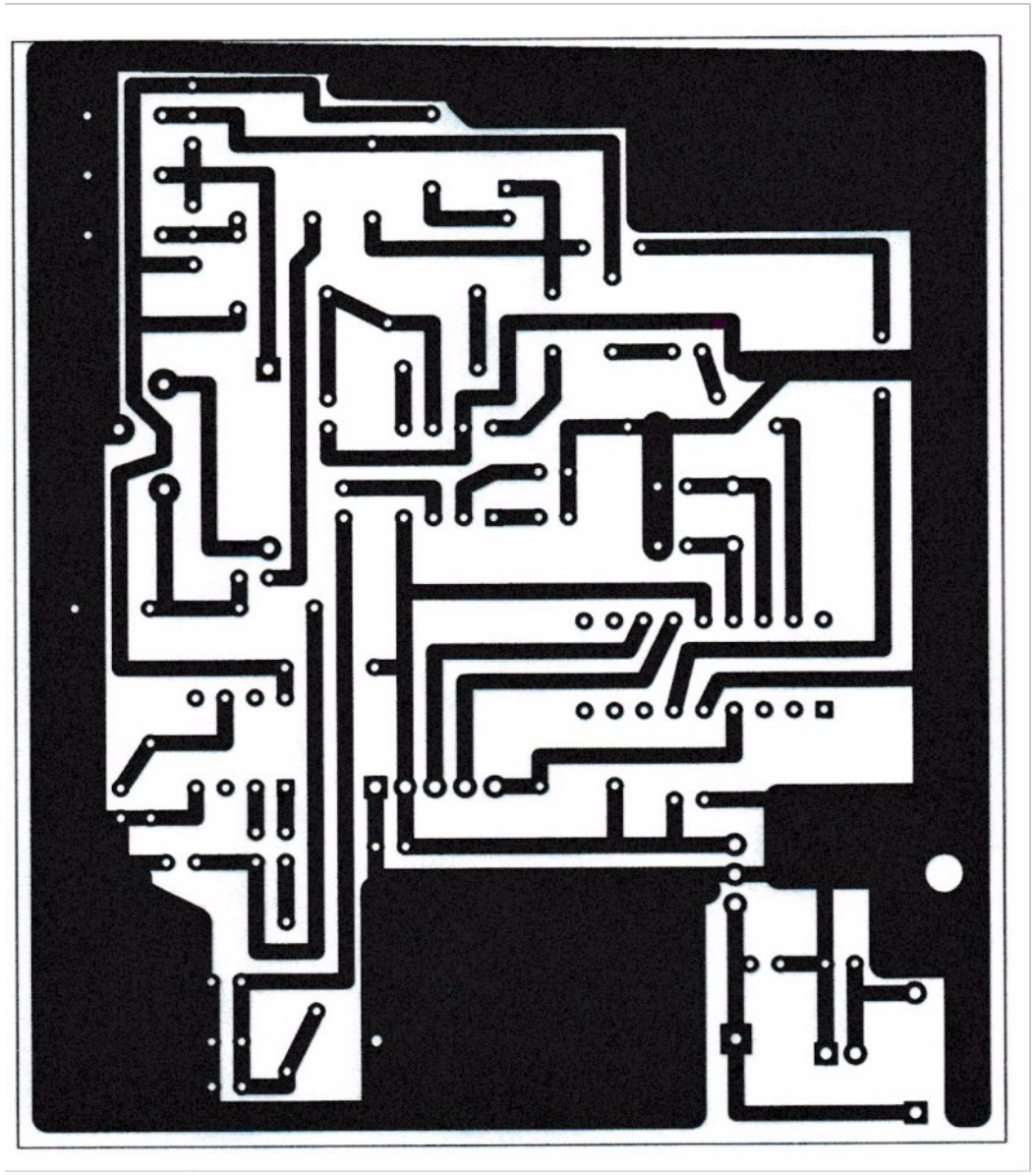
Circuit Diagram of the Automatic Blue Screen Eliminator



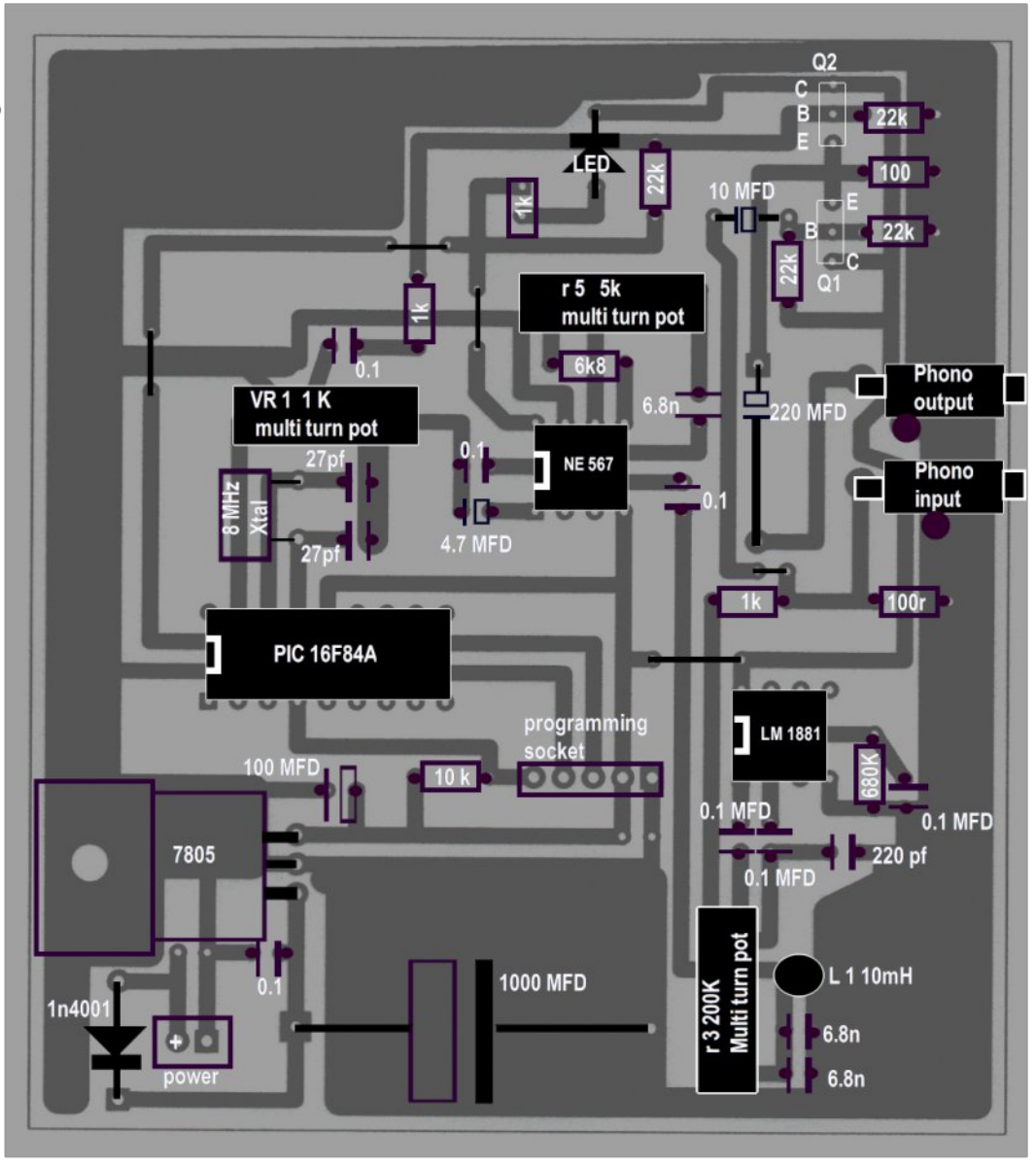
(Note pin 8 on the LM567 switches on and off the bias to Q2 SYNC/NOISE adder)

Finally, with a weak signal, adjust R3 so as to back it off a bit.

The PIC software generates fully interlaced SYNC and is also turned on and off with the SYNC DETECTOR circuit fed to PIC pin 6.



PCB Component side



PCB Copper View

DATV-Express Project - March update report

By Ken W6HHC

Art WA8RMC reported that the inventory of DATV-Express boards is in fine shape. Some of the boards shipped to Europe in January appear to have taken a long time for delivery and two boards apparently never did made it? Art is working with those two customers by sending replace boards to them.

Ken W6HHC has updated the NOTES.txt file (aka README) on the *http://www.DATV-Express.com* web site to describe three known problems in v1.23 of the Express_DVB-S_Transmitter software for Windows. These three known problems are:

- *Windows-for-Italian OS does not select capture-camera properly because of use of a "Italian-friendly" setting name. Using vMix should be a work-around.*
- *Windows-for-Japanese OS does not select capture-camera properly because of use of a "Japanese-friendly" setting name. Using vMix should be a work-around.*
- *The MER readings using MiniTiouner receiver for DVB-S2 transmissions appear to read "lower" in value than when using the DVB-S protocol. Due to an "unknown reason", the 32APSK modulation will not decode on a SatLink Analyzer unless the DATV-Express software has modified the 90 symbol preamble magnitude on the DVB-S2 frame. The received DVB-S2 signals are still very robust.*

This new NOTES file can be downloaded from the DOWNLOADS page.

Ken W6HHC reports that he has started to test the Portsdown Project from BATC with a configuration set-up of:

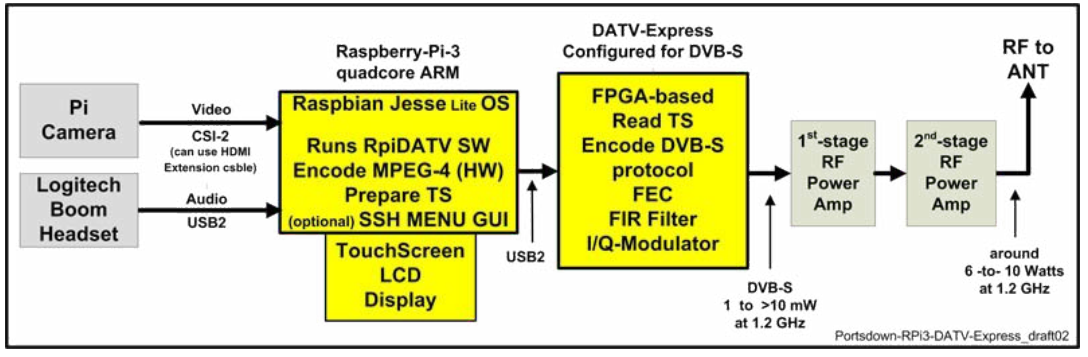
1. *Raspberry-Pi-3 with LCD Touchscreen display*
2. *BATC-customized RpiDATV software from Evariste F5OEO*
3. *A DATV-Express board as the modulator and frequency selector*

The LCD Touchscreen is the normal GUI.

Charles is continuing take a break from the DATV-Express efforts and is involved in two long-term efforts:

1. *A very complex Digital-Pre-Distortion research effort that will use CUDA video board to perform high-speed math and initially use the LimeSDR board to transmit.*
2. *Has become part of the team working on the USA Phase 4B geosync satellite project that will use DVB-S2X protocol primarily for data transmission of voice, JPEGs and TCP-IP.*

No new major releases of DATV-Express (even alpha) are expected to be available until the launch of Es'hail 2 sat in 2018.



Block Diagram of a Typical Portsdown set-up for using the DATV-Express board for DVB-S modulator

"Project speed set to slow"....de Ken W6HHC

ATV reception via HAMNET and RaspberryPi (Christof, OE8BCK)

In January 2017 a new version 7.0 of OpenELEC with KODI 16.1 has been released. This enables all radio amateurs with HAMNET access to build up a receiver for digital ATV streams using a Raspberry Pi3 and an HDMI video monitor.

Explanation

OpenELEC is a software distribution for simple installations of media player *KODI*. This should form a small remote-controllable media center for pictures, music and videos.

Live streaming from satellite TV receivers or from online sources like amazon and netflix has been added recently.

Hardware

My own solution comprises of Raspberry Pi3 with power unit, housing and SD card, preferably more than 8 GB class 10 with UHS-1 from big producers for the sake of data rate. A decent power unit should be used in order to prevent crashing.

For monitoring I am using a TV monitor with HDMI port out of my shack. Maybe KODI is also available for Windows PC and for Android units, but I am used to Linux.

Installation

After downloading the *OpenELEC image data* for Raspberry Pi2, which is the same as for version 3, it is put onto the SD card for booting the minicomputer. The installation routine there is asking for UI language etc., then KODI pops up and needs configuration of media sources and sinks.

ATV streams as video sources

Now we need HAMNET access for Raspberry Pi, which is connected to my central web router. This router sends HAMNET addresses (44.x.x.x) with network mask 255.0.0.0 to my HAMNET antenna/router “NanoBeam” configured as ?Network Mode: Router”. In Raspberry Pi the HAMNET address file name ends with “.strm”, i.e. for my nearest node Khuenegg the file name is “khuenegg.strm” with the address line inside “rtsp://44.143.145.186:5131/0”

That seems simple, but you should know Linux usage. I am usually taking SSH on RasPi and call text editor “nano” writing “nano Videos/Khuenegg.strm”. Then the “control” button and the icons at the lower screen border are used.

It is advised to make a subdirectory “Videos” for such media files. Then the ATV-Streams are added as sources by “Videos - Files - Add Videos”. With clicking “browse” you choose the “Home folder” and there “Videos”. By clicking “ok” an alternative name for “Videos” can be chosen, which will show

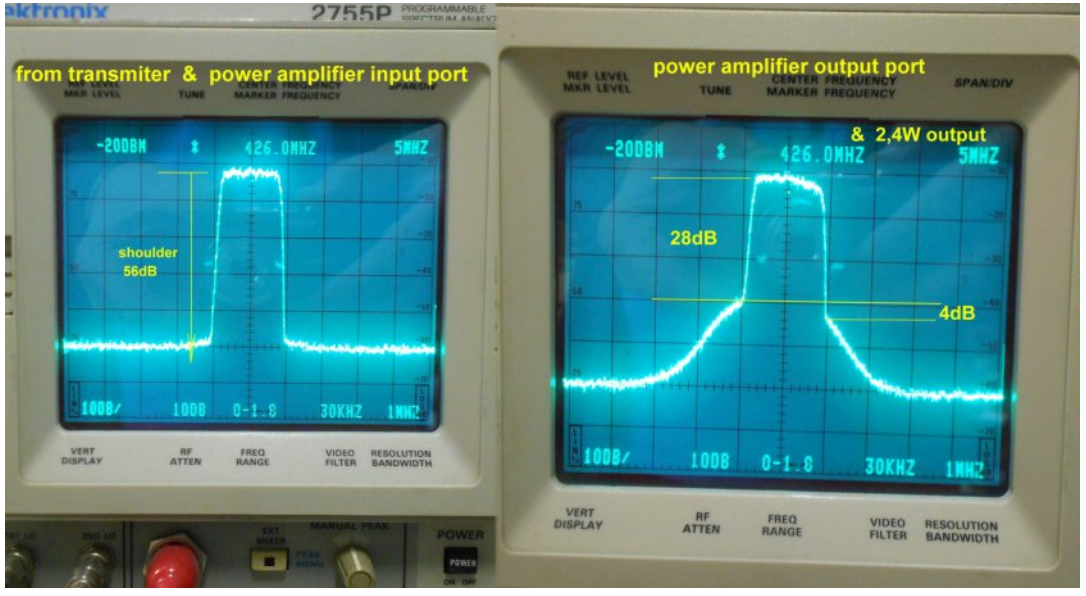
all files in this directory. One more "ok" is defining in the box above left, that this is containing videos.

If you click on the subdirectory now, it will show all files in it, and with clicking on one of it, the concerned ATV stream begins running.

This procedure is only large at the start, afterwards the "eco system" KODI/OpenELEC is a nice platform to view AV sources or DATV via HAMNET. There is a huge user base producing more add-ons like weather forecasts or i.e. satellite TV, thanks to KODI.

DVB practice (Darko, OE7DBH)

When I started to construct a DVB PA using the old M series or the RA series by Mitsubishi with plastic housing, I found bad linearity and high shoulders compared to the new RA series with metal housing! Please avoid the older versions in devices for digital ATV!

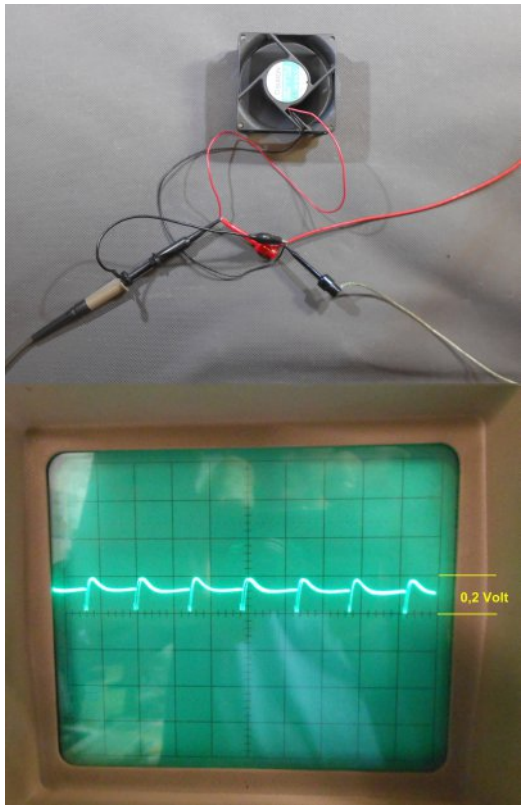


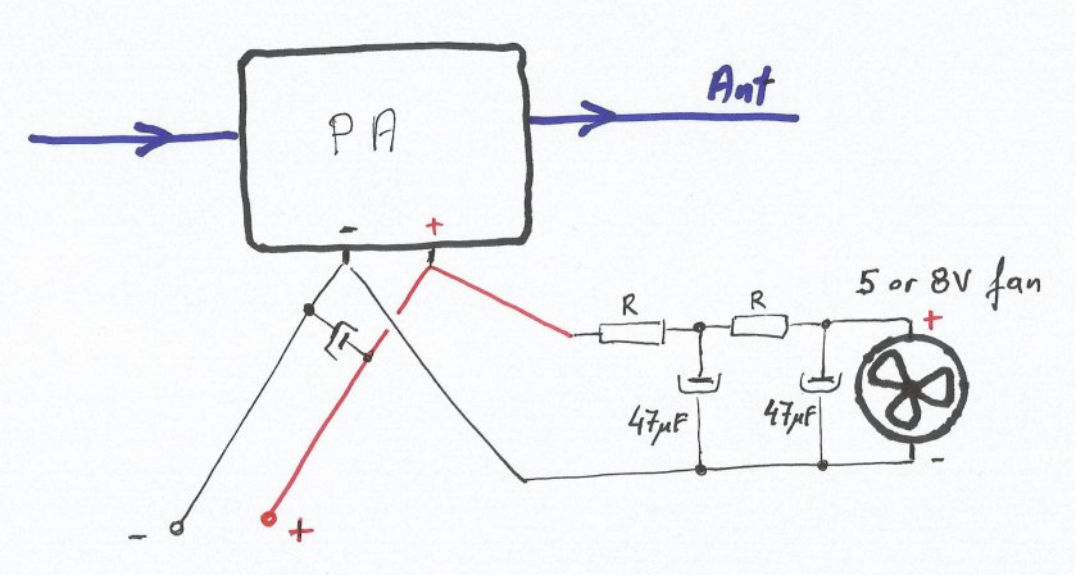
Left hand side metal, right hand side plastic housing

In home made devices with PA normally the cooling fan is fed by the PA power supply, but be warned:

Most fans are feeding back a noise signal into the power rail which is disturbing the transmitted rf signal.

In order to prevent that a filtering circuit like the one in the diagram below is advised.





DVB-T-AGC for repeaters (Henry, F4WBG)

On "ebay" I found an interesting device ready for use with SMA sockets: central part is the IC AD 8367, a VCA or AGC amplifier for up to 500 MHz.

In AGC mode it is producing a constant output signal at -12,5 dBm with a DVB-T source signal variing from -45 dBm to 0 dBm. In a transponder installation without demodulation and re-modulation I am using the AD 8367 for AGC with 45 dB dynamic. Receiving on 1267 MHz a very good bandpass filter for 70 MHz IF is feeding the AGC IC. Mixer output is on 550 MHz fed by a PLL-VCO by DF9NP for 9750 MHz in order to generate 10,3 GHz transmitting output.

I constructed this device as I was not able to receive a DATV repeater directly which is 125 km away from my QRA. On a hill about 40 km away in the same direction a 2m phone repeater was line of sight for both. The repeater club allowed me to install my transponder on their antenna mast. A first version without AGC device overloaded the output on 10,3 GHz sometimes, so my receiving satellite TV receiver did not work properly. After adding the AGC a now stable MER solved this problem.

AGC device on ebay:

http://www.ebay.de/itm/331710882190?_trksid=p2055119.m1438.l2649&ssPageName=STRK%3AMEBIDX%3AIT

Documentation:

http://www.analog.com/media/en/technical-documentation/evaluation-documentation/AD8367.pdf

https://www.oe7forum.at/

Translations by Klaus, DL4KCK

TV Amateur is a German Language ATV Magazine It is published 4 times a year and if you would like to subscribe go to *http://agaf-ev.org/* NOTE new URL

External links

If you have an eBook reader that does not have WiFi then you will not be able to use the hyper-links in this publication. If you have an eBook reader that has WiFi then you will be able to providing you are in a WiFi zone.

But if you have a Kindle 3G then yes, but only to Amazon, and there is not a lot of ATV material on their site.
Smart phone reading apps are ok providing that you have a 3G data connection.

Note: These links will fire up your devices browser and if you are using 3G/4G then you will incur data usages charges.

Legal Niceties (the small print)

E&OE. Whilst every care is taken in the production of this publication, dotMOBI accepts no legal responsibility for the advice, data and opinions expressed. dotMOBI neither endorses nor is it responsible for the content of advertisements or the activities of those advertisers. No guarantee of accuracy is implied or given for the material herein. dotMOBI expressly disclaims all liability to any person in respect of anything and in respect of the consequences of anything done or omitted to be done wholly or partly in reliance upon the whole or any part of this publication.
As the regulations for the operation of radio frequency equipment vary in different countries, readers are advised to check that building or operating any piece of equipment described in dotMOBI will not contravene the rules that apply in their own country.
All copyrights and trademarks mentioned in this publication are acknowledged and no infringement of the intellectual copyright of others is intended.

Copyright

Notice to Contributors

Authors are alone responsible for the content of their articles, including factual and legal accuracy, and opinions expressed by them may not reflect the editorial stance of the publication. Material submitted to dotMOBI should not infringe the copyright of other writers or bodies.
Contributions are accepted for publication on this basis alone.
dotMOBI publications - *http://cq-datv.mobi*

Author Guidelines

CQ-DATV welcomes contributions from our readers. It does not necessarily have to be specifically ATV, as long as it is of interest to our readers.

Although a formatted article showing the layout can be sent, we prefer an unformatted text file of the script, along with annotations of where important images should be placed. All images should be identified as Fig 1 etc and sent seperately.

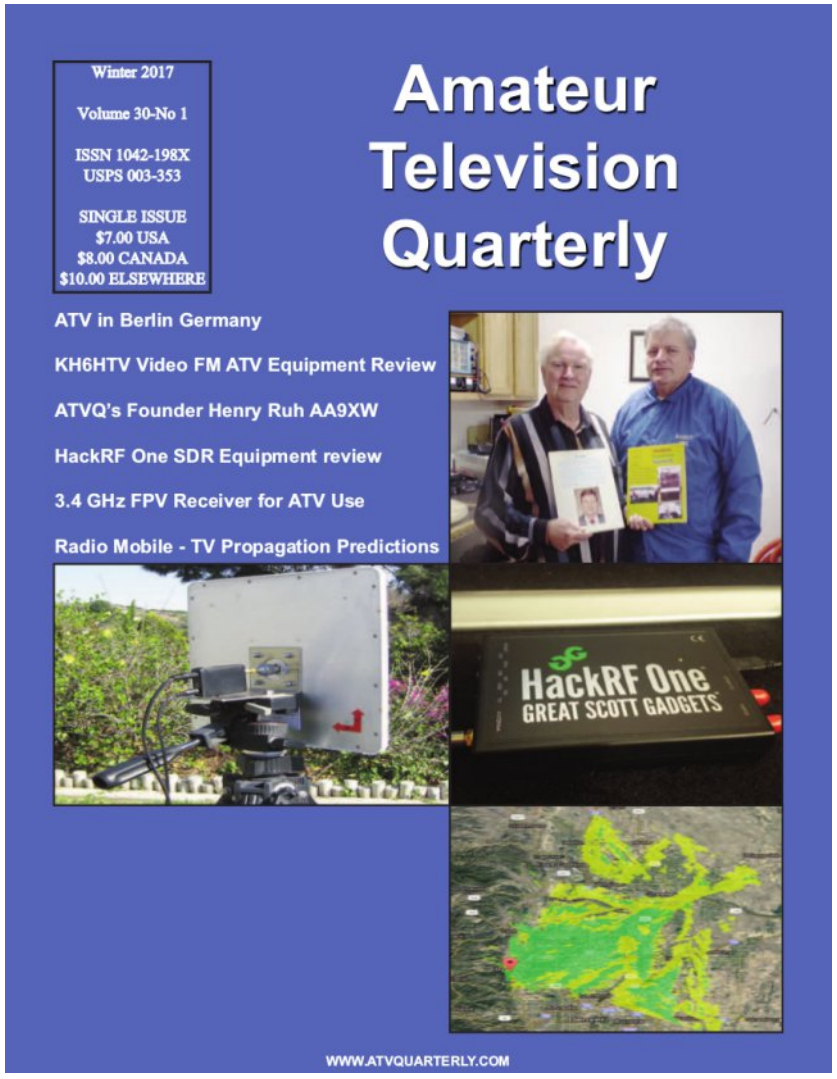
Images should be in PNG format if possible and the best quality available. Do not resize or compress images, we will do all the rework necessary to publish them.

If you are sending a construction project, please include the dimensions of any pcb's and make the pcb image black and white, not greyscale.

CQ-DATV reserves the right to redraw any schematics and pcb layouts to meet our standards.

Coming up in CQ-DATV

Is this the latest issue of CQ-DATV? *Click here* to go to our web site to check to see if there is a later edition available.



ATV Quarterly - Don't miss another issue! Subscribe Today

USA $24.00 year, Canada/Mexico $27.00 year DX $37.00 year (US $) Cyber: $15/yr. Visa, M/C, AMEX, PayPal via Internet: www.atvquarterly.com
Cheques or Money Orders to P.O.Box 1594 Crestline CA 92325
Published by ATV Quarterly tel (909) 338-6887
email: wa6svt@atvquarterly.com