

# TV TUNERİ İLE RADYO ALICISI

TA7AKA – Murat KAYA

*Eskiden pek çok amatör elektronikçiyi caydıracak kadar karmaşık devreler kurmayı gerektiren sistemler teknolojiye ilerlemelerle birlikte öylesine basitleşti ki... Daha önce yüzlerce parçanın bir arada uyumlu bir şekilde çalışmak zorunda olduğu ve onlarca ayar gerektiren devreler artık tümleşik devre teknolojisi sayesinde hem kolayca yapılabiliyor, hem eskisi kadar ayar gerektirmiyor. Bununla birlikte geçmişteki devrelerde hayal bile edilemeyecek ek özelliklere de sahip olabiliyor.*

İşte bu uygulamalardan bir tanesi de birazdan anlatacağımız TV tuneri, basit bir IF katı ve bir mikrodenetleyiciden oluşan radyo alıcısı... Bu alıcı ile 45MHz ve 860MHz arası kesintisiz bir şekilde 50KHz adımlarla taranabiliyor. Üstelik hassasiyeti de gayet güzel.

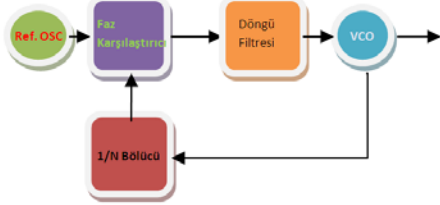
Aslında fikir kesinlikle yeni değil. Uzun zamandır buna benzer uygulamalar PLL içermeyen tunerlerle de yapılıyordu, frekans kapsamında bir değişiklik olmuyordu, sadece ayarlama işlemi eskilerden hatırlayacağımız çok turlu potansiyometrelerle yapılıyordu. İnternette buna benzer onlarca çalışma var.

Çalışmada çok yeni bir şeyler yaptım demek sahtekârlık olur. Kullandığım bileşenlerin veri sayfalarında zaten uygulama şemaları var, ben bunların üzerinde çok az bir değişiklik yaptım. Belki en büyük katkı olabilecek kısım programlama...

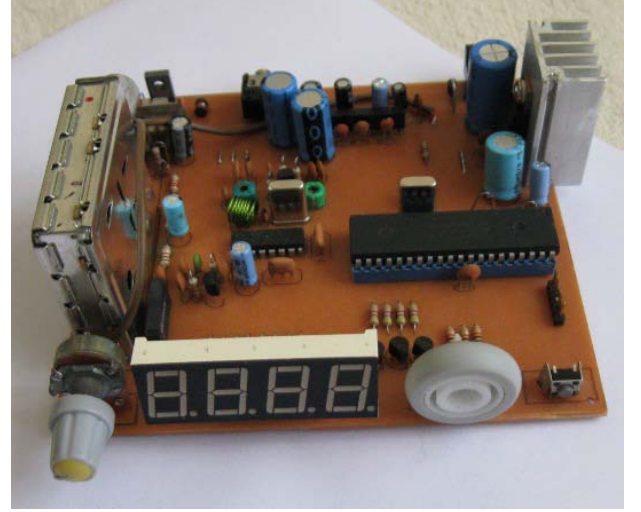
## PLL Hakkında...

Yukarıda PLL adında bir şeyden bahsettim. PLL (Phase Locked Loop - Faz Kilitli Döngü) konusu özellikle yeni başlayanlar için can sıkıcı veya zor gelebilir, ancak radyo sistemlerinin bugün için olmazsa olmazları arasına giren bu sistemi anlamak bize çok büyük avantajlar sağlayacaktır.

PLL sistemini burada detaylı bir şekilde verebilmek mümkün değil, zaten konumuz da o değil. Ancak özetlersek sistem şu şekilde çalışıyor:



Faz karşılaştırıcı referans osilatörün – ki bu osilatör son derece kararlı bir osilatördür, örneğin kristal osilatör – sinyali ile 1/N bölücüden gelen sinyalleri karşılaştırır ve bir fark gerilimi üretir. Bu fark gerilimi döngü filtresinde süzülerek Voltaj Kontrollü Osilatöre (VCO – Voltage Controlled Oscillator) iletilir. VCO bu gerilimle orantılı bir frekansta sinyal üretir. Faz karşılaştırıcıda referans osilatör ile karşılaştırılan sinyal işte bu VCO sinyalinin N sayısına bölünen miktarıdır. Bu durumda VCO çıkışı referans osilatör frekansının N katına eşit olur. Eğer VCO frekansı değişirse faz karşılaştırıcının çıkışındaki gerilim değişecek ve VCO'yu istenen frekansa tekrar getirecektir. N değeri değiştirilerek VCO referans frekansın herhangi bir tamsayı katına kilitlenebilir.



Yukarıda da görüldüğü gibi devre üzerinde çok fazla eleman yok. Bu nedenle bize sıkıntı çıkaracak pek bir yer de olmuyor.

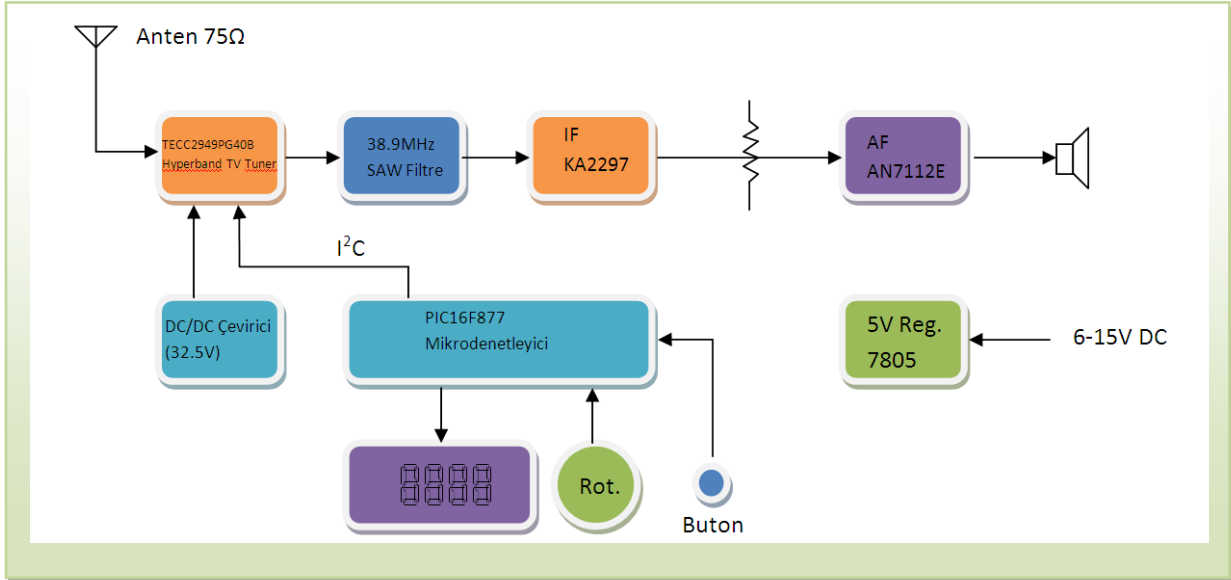
Sözü çok fazla uzatmadan devremizi tanımaya başlayalım.

## İşlevsel Blok Şeması

Esas devre şemamıza geçmeden önce radyo alıcımızın işlevsel bloklarını gösteren şemamıza bakalım. Bu şema hem radyomuzdaki birimler arası işleyişi gösterecek hem de devremizi daha kolay anlamamızı sağlayacak.

Şemada da görüldüğü gibi antenden gelen sinyaller doğrudan tuner katına geliyor. Burada yapılan işlemlere daha sonra bakacağız. Ardından 38.9MHz SAW filtresinden geçen ara frekans sinyali KA2297 entegresinden oluşan IF katına geliyor. Bu entegre ucuz radyolarda bulunabilen bir entegre – en azından ben bunu öyle bir radyodan sökmüştüm ☺ - . Ardından yine benzer şekilde bulunabilecek olan 0.5W çıkışlı AN7112E entegresinde ses sinyalleri yükseltilerek hoparlöre veriliyor.

PIC16F877 mikrodenetleyicisi burada tunere frekans bilgisini yollama, rotary anahtardan ve butondan gelen verileri yorumlama ve frekans bilgisini LED ekranda gösterme işlerinde kullanılıyor.



Devremiz 5V DC ile çalışıyor. Bu gerilimi bir 7805 regülatörü kullanarak elde ediyoruz. Giriş gerilimi geniş bir aralık için sorun çıkarmadan çalışıyor. Yine de 12V seviyesini geçememek gerek, en azından uzun süreli kesintisiz bir çalışma yapılacaksa en çok 12V iyi bir değer.

Tuner katı varikap diyotlar için yaklaşık 33V seviyesinde bir DC gerilime ihtiyaç duyuyor. Bunu pek çok şekilde elde edebiliriz. Ben burada özellikle kuş sesli kapı zillerinde kullanılan türden küçük bir ses çıkış transformatörü kullandım. 5V gerilimi çok iyi bir şekilde yükseltiyor. Zener nedeniyle tuner gerilimi 32.5V seviyesinde kalsa da bu sorun değil. Bu değer de işimizi görmeye fazlasıyla yeter.

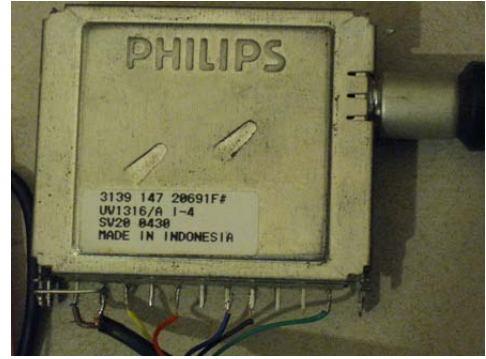
Şimdi bu bileşenleri detaylı bir şekilde açıklayalım.

### Tuner Katı

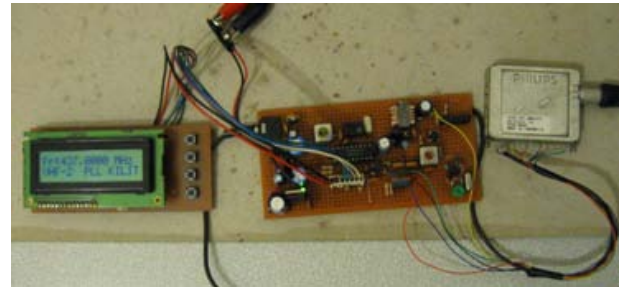
Bu küçük metal kutu aslında benim bu devreyi yapmaya başlamamdaki en büyük etken diyebilirim. Bu tuneri okul müdürümüzün laboratuvarımıza bağışladığı bozuk televizyondan aldım ☺. Tunerin adı sizi korkutmasın, çoğu televizyondaki tunerler burada işimizi görür.

İlk elime geçen tuner Philips marka UV1316 tuneri idi. Hemen hepsi benzer standartlara sahip ve birbirleri yerine kullanılabilir. Bende buradaki Samsung TECC2949PG40B'den başka bir Philips UV1316 ve bir de Panasonic ENV57K04G3RF adlı tunerler var. Bunların hepsi de birbirleriyle uyumlu ve sanırım sizin elinizde de benzer bir şeyler varsa uyumlu olma ihtimali çok yüksek.

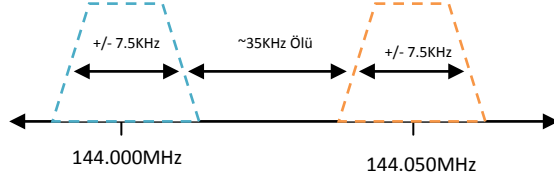
Tunerlere ait veri sayfası bulmak gerçekten zor. Bundan 4-5 yıl kadar önce tuner kullanarak ilk devremi yaparken internetten UV1316 tunerine ait bir veri sayfası bulup indirmiştim. Tunere ait programı yazarken de bu veri sayfasına güvenerek yazdım ve ilk deneme tam bir hüsrana oldu.



Bir kodlara bir veri sayfasına bakarak saatler geçirdim ama hata bulamadım. Tek çare kalmıştı; tuneri açıp içeride neyin yanlış gittiğine bakmak... Tuneri açtığımda sorunun ne olduğu anlaşıldı, aslında UV1316 modernize edilmişti ve içinde ayrı iki entegre halinde olması gereken PLL ve Mixer-Osilatör katları tek entegrede birleştirilmişti. Entegrenin adını okuduğumda sorun da çözülmüştü; TDA6508A. Bu entegrenin veri sayfasını indirip kodları yeniden düzenledim ve devre gayet güzel bir şekilde çalışmaya başladı. Üstelik UV1316 için zorunlu görünen ve frekans değerinde eksantrik sayılar çıkmasına neden olan 62.5KHz'lik adımlardan da kurtulmuş oldum çünkü bu entegre 31.25, 50 ve 62,5 KHz'lik adımlardan istediğimi seçmeme izin veriyordu. Ben de 50KHz'i seçmiştim. İşte o devre...



O zamanki devrem biraz dağınık olmuştu ama öyle durduğuna bakmayın, gayet güzel çalışıyordu. Belki tek kötü tarafı, 50KHz adımlarla ilerlemesine rağmen 15KHz bant genişliği olması ve bu nedenle de adeta yerdeki çamurlu suya basmamak için sekerek ilerleyen birinin basmadığı yerler gibi dinlenemeyen bir sürü boşluk bırakmasıydı.



Bu sıkıntıdan kurtulmanın iki yolu var; ya adım aralıklarınızı küçülteceksiniz –ki bu iş ortalığı biraz daha karıştırmak demek – ya da bant genişliğinizi yüksek tutacaksınız. Ben şimdilik basit olan ikinci yolu seçtim ama amatör radyoculuk adına çok iyi bir fikir değil. Nedenini yazımın devamında anlatacağım.

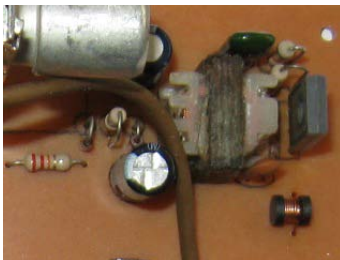
Bu arada ilk ara frekans (Intermediate Frequency - IF) katımız tunerde olacak. Tuner çıkış filtresinin geçiş bandının tepe noktası yaklaşık olarak 37MHz. Ben de bu frekansa olabildiğince yakın olacak şekilde 37.7MHz değerini kullandım. Neden 37.7MHz? Bu da yazımın devamında açıklayacağım.

Frekans ayarı bu tunerlerde sayısal veriler kullanılarak yapılıyor. Bu veriler IC veri yolu üzerinden tunere gönderiliyor. Bu yolun nasıl çalıştığını burada anlatmayacağım, internette bu konuda bolca bilgi var. Ben sadece yüzeysel olarak tunerde hangi verilerin hangi kaydedicilere gönderilmesi gerektiğini yazacağım, ama mikrodenetleyici ile ilgili açıklamalarda...

Tunerimiz frekans ayarı için harici bir yüksek gerilim kaynağına ihtiyaç gösteriyor. Şimdi buna bakalım.

### DC/DC Çevirici Katı

Tuner varikap diyotları sürmek için yaklaşık 33V gerilime ihtiyaç duyuyor. Veri sayfasında bu gerilim için verilen sınır değerler 30V – 35V arasında. Bu gerilim çok çıkışlı bir besleme transformatörü kullanılarak doğrudan elde edilebileceği gibi herhangi bir buck-booster devresi hatta 555 gibi entegre devrelerle bile kolaylıkla üretilebilir. Benim burada kullandığım ise basit tickler osilatörü tarzında düzenlediğim bir devre.



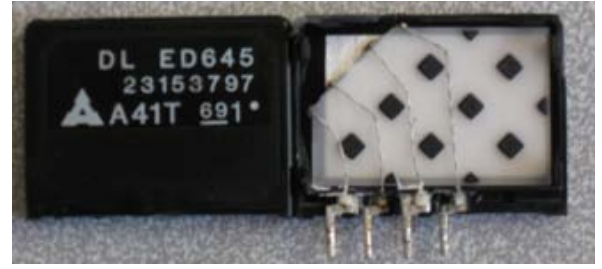
Görüldüğü gibi son derece basit ve 5V gerilimi kolayca 32.5V'a kadar yükseltiyor. Aslında yaklaşık 40V'a kadar çıkıyor ama zener ile basit bir regülasyon yapıyoruz ve gayet sağlıklı bir tuner gerilimimiz oluyor.

Aslında bu da bir osilatör ve radyo devrelerinde açıkta osilatör bırakmak enterferans nedeniyle pek mantıklı değildir, ama bu bir deneme devresi ve amaç yapılanları gösterebilmek. Yine de devreyi yapmak isteyenler bu bölümü küçük bir metal kapakla kapatabilirler. Baskı devreyi buna uygun hazırladım.

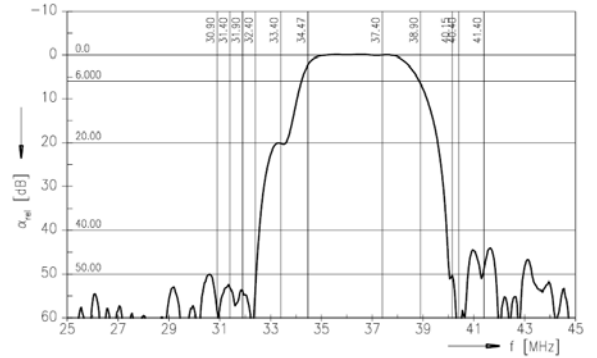
### SAW Filtre

SAW (Surface Acoustic Wave – Yüzey Akustik Dalga) filtreleri, istenilen frekansta ve istenilen bant genişliğinde sinyal geçirebilen ve bunların dışındakileri de çok iyi bir şekilde bastırabilen bir çeşit kristal yapıdır. Bu filtreyi bulmak çok kolay, televizyon tamircilerinde bile bulunur. Ama en iyisi tuneri söktüğünüz devreden bunu da alın.

Bu filtrelerin bant genişlikleri yaklaşık 6-7 MHz civarında. Aşağıda bu amaçla eskiden kullanılan bir SAW filtrenin içini görüyorsunuz.



Bu filtreler ara frekans katları için gerçekten biçilmiş kaftandırlar. Aşağıda EPCOS G1962M SAW filtresi için frekans cevap eğrisi görülmektedir.



### Ara Frekans (IF) Katı

Süperheterodin sistemlerde yer alan çok önemli katlardan biri de ara frekans katıdır. Bu kat belli, sabit bir frekansa ayarlanmış bir radyo alıcısı gibi düşünülebilir. Bir alıcıya ait bant genişliği, kazanç gibi özellikler birinci derecede bu kata bağlıdır.

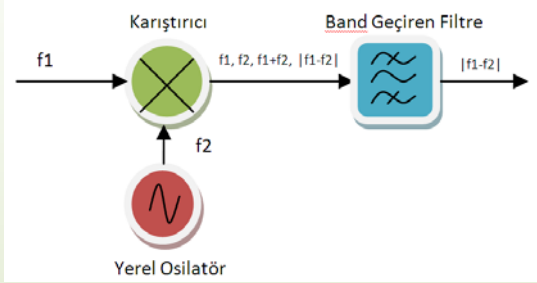


Ayrıca radyo alıcısının demodülasyon türünü de bu kat belirler.

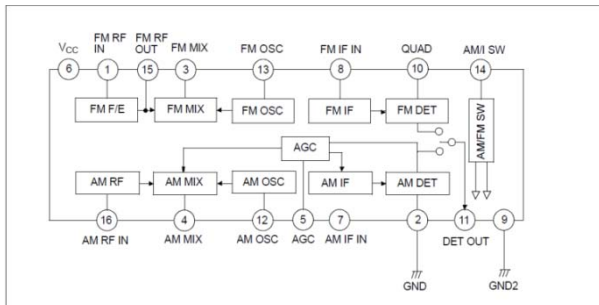
Bizim burada IF olarak kullandığımız basit bir AM/FM radyo entegresi olan KA2297. Eğer bunu bulamazsanız bu entegrenin karşılığı olan TA2003'ü de kullanabilirsiniz.

### Süperheterodin Nedir?

Süperheterodin şu anda çoğu radyoda, LNB'lerde ve diğer pek çok yüksek frekans uygulamasında kullanılan bir yöntemdir. Bu teknikte iki frekans sinyali karıştırıcı (mixer) adı verilen bir katta birbirleriyle "çarılır". Bu işlem sonucunda karıştırıcı çıkışında 4 sinyal görünür : 1. Sinyal, 2. Sinyal, her ikisinin toplam frekansına sahip bir sinyal ve iki sinyalin frekanslarının farkı frekanstaki 4. Sinyal (dengeli karıştırıcı adı verilen bir karıştırıcı türünde çıkışta sadece toplam ve fark frekanslarına sahip sinyaller görülür, giren sinyaller görülmez).



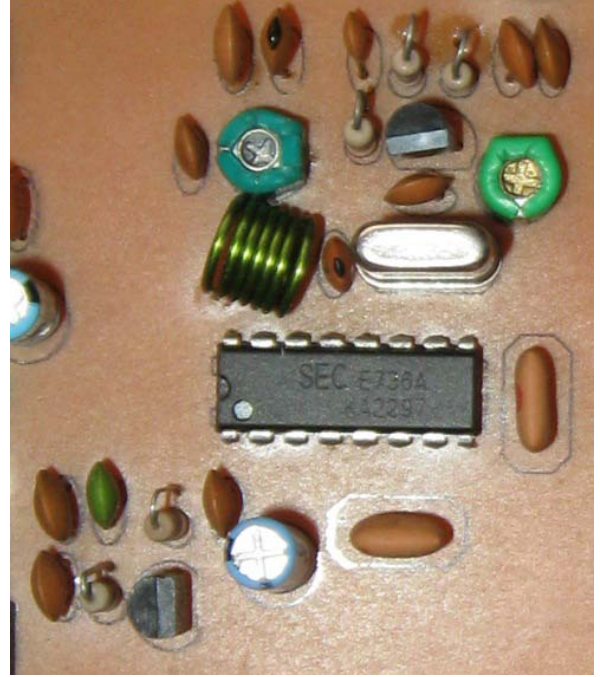
Karıştırıcının ardından gelen bir bant geçiren filtre f1 ve f2 frekanslarının farkını geçirecek şekilde ayarlanır ve sistemin çıkışında sadece bu fark sinyali gözükür. (Fark sinyalini mutlak değer içinde gösterdim, çünkü negatif frekans olmaz.) Yerel osilatörün frekansını (f2) değiştirebiliyorsanız giriş frekansını da (f1) seçebilirsiniz demektir. Sonuçta elde edilecek fark frekansı sabit olacağı için üzerinde işlem yapmamız gereken bir tane frekans var demektir, buna Ara Frekans (Intermediate Frequency - IF) denir ve bundan sonraki devreyi bu frekansa bir kez ayarladıktan sonra işimizin çoğunu yapmış oluruz. Artık tek yapmak gereken yerel osilatörün frekansını değiştirmek. Örneğin evlerimizdeki FM radyolarda IF 10.7MHz'dir. Siz ister 88MHz'i dinleyin isterse 108MHz'i alıcı bunu önce 10.7MHz'e çevirir ve öyle işler.



Entegre devrenin blok diyagramından da görülebileceği gibi KA2297 hem FM hem de AM sinyallerini alabilecek şekilde yapılmıştır. Ancak

AM kısmı bu devre için kullanılmayacak. Sadece FM kısmıyla ilgileneceğiz.

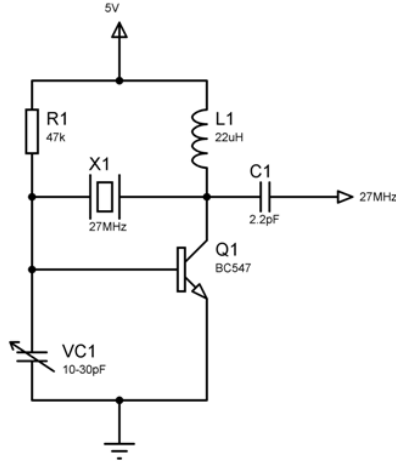
Bu entegre birçok FM alıcı entegresinden farklı olarak FM ön/son (front/end) katlarına da sahip. Bu da 37.7MHz'lik 1. IF frekansını 10.7MHz olan 2. IF frekansına çevirirken çok işimize yarayacak.



IF katı bundan ibaret diyebiliriz. Yeşil bobin ve hemen üzerindeki 20-30pF ayarlı kondansatör 37.7MHz'lik frekansa ayarlı ve KA2297'nin FM RF yükseltici/seçicisine bağlı. Devreyi çalıştırdıktan sonra bu kısmın ayarını yapmak çok kolay olacak.

Şeklin altında elektrolitik kondansatörün sağında yer alan 10.7MHz'lik filtre 2. IF katına ait. Entegrenin sağındaki ise yine 10.7MHz'lik diskriminatör rezonatörü.

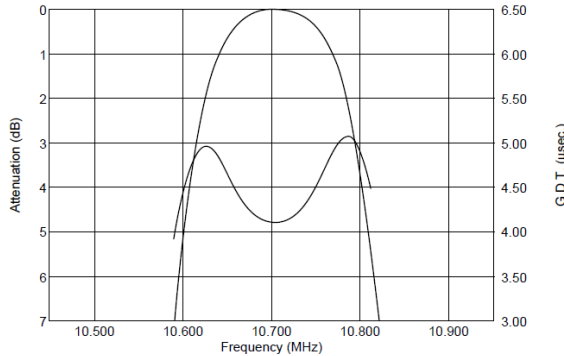
Gelelim entegrenin yukarısında yer alan kristalli osilatöre. Aslında bu kadar karmaşık olmasına gerek yoktu, çünkü entegre kristal osilatörü destekliyor. Yani bir kristali entegrenin 13 numaralı pini ile Vcc ya da GND arasına bağladığımızda osilatörün çalışması gerek. Ama benim kullandığım 27MHz kristal ile olmadı. Nedenini anlayamadım, belki de kristal kesimleri ile ilgilidir. Çünkü yine bu kristal gibi 3. Harmonik frekansında çalışan 26.603MHz'lik kristallerle gayet güzel çalıştı. Ben de risk almamak için bir osilatör katı ekledim. Osilatör tasarlarken de eğer kristal uygunsa ek osilatörü kurmadan bağlantı yapacak şekilde bir yol izledim. Aksi takdirde aşağıdaki gibi bir osilatör 3. harmonikle çalışan kristaller için çok daha uygundur.



Osilatör konusundaki tavsiyem önce kristali takıp deneyin. Bunun için kristali yerine takın ve sağdaki ayarlı kondansatör yerine atlatma (kısa devre köprüsü) yapın. Olmazsa osilatörü kurun.

Osilatörle ilgili olarak söylemek istediğim bir diğer konu ise yine enterferans yani girişim. Bu osilatör açıkta olduğu sürece 27MHz'lik sinyaller yaymakta bu da harmonikleri olan 54, 81, 108,... MHz frekanslarında sahte sinyaller alınmasına neden oluyor. Bunun çözümü yine metal kutu, ya da diğer adıyla Faraday Kafesi.

Alicımızın bant genişliğini belirleyen kısmın da burası olduğunu söylemiştik. Burada bir tane bant filtresi kullanılıyor. Buradaki Murata SFE10.7MS2-A filtresi yaklaşık 230KHz bant genişliğine sahip. Aşağıda bu filtreye ait frekans tepkisi eğrisi görülüyor.



Bant genişliği bu kadar fazla olunca bunun bazı avantajları ve dezavantajları oluşuyor. Avantaj olarak il söyleyebileceğim 50KHz'lik tarama sırasında dinlenilmeyen frekans – ölü bölgeler – bırakmaması ve eğer sayısal veri aktaracaksanız dar bantlı bir alıcıya göre daha yüksek veri hızına sahip olabilmesi. Bir de avantajdan sayarsanız FM radyo yayınlarını ve karasal televizyon yayınlarının seslerini dinleyebilirsiniz. Dezavantajlara gelince; ilk olarak bu radyo ile özellikle amatörlerin kullandıkları ve 2m ve 70cm bantlarında çalışan amatör telsiz yayınlarını dinleyemezsiniz. Bu yayınlardaki frekans değişimi bu alıcıya göre o derece azdır ki, yayınları

radyonun arka plan gürültüsünde bir azalma ya da kaybolma olarak fark edersiniz. Ayrıca bant genişliğinin çok olası kazancı azaltırken gürültüyü de artırır, yani uzak veya zayıf sinyallerde sorun yaşarsınız. Amatör frekanslarda birkaç kanal bu alıcının bantına sığacağından aynı anda birkaç kanaldan gelen sinyalleri alma şansınız(!) olur (örneğin PMR telsizlerin 8 kanalını bu alıcıda 446.050MHz üzerinden alabilirsiniz, tabii ki kimin hangi frekansta olduğunu anlama şansınız yok ve zaten sesleri de duyamıyorsunuz). NOAA uydularından gelen APT resim sinyallerini alsa da yine bol gürültü nedeniyle işe yarayan bir resim elde etme şansınız çok az.

Tüm bu dezavantajları ortadan kaldırmanın yolu iki ayrı IF katı kullanmak. Bu şekilde hem geniş bantın avantajlarından vazgeçmemiş oluyoruz hem de dar bantlı yayınlardan da fedakârlık etmiyoruz. Eğer, yok geniş bant bana lazım değil ben dar bantlı yayınları dinlesem bana yeter dersiniz o zaman IF katını MC3359, MC3361 veya MC3371 gibi bir entegre kullanarak değiştirmeniz gerek. Tabii baskı devrede de buna uygun değişiklikler yapmalısınız. Yine bu ihtimali düşünerek IF katını da derli toplu yaptım. İsterseniz ARES programını kullanarak istediğiniz şekilde değiştirebilirsiniz. Ben bundan sonraki uygulamamda hem geniş bant FM (WFM – Wideband FM), hem dar bant FM (NFM – Narrowband FM), hem de genlik modülasyonu (AM – Amplitude Modulation) için 3 ayrı IF katı yapmayı düşünüyorum.

Peki, her şey iyi güzel de 1. IF frekansı neden 37MHz değil de 37.7MHz? Kolayca yapılabilecek bir hesapla gösterelim. 1. IF ten gelen sinyaller 27MHz'lik yerel osilatör ile karıştırılıyor ve bu iki sinyalin toplamı ve farkları elde ediliyor. Toplamı kullanmıyorum çünkü;

$$37.7+27=64.7\text{MHz}$$

bu kadar yüksek frekans bana lazım değil. Bir de farklarına bakalım;

$$37.7-27=10.7\text{MHz}$$

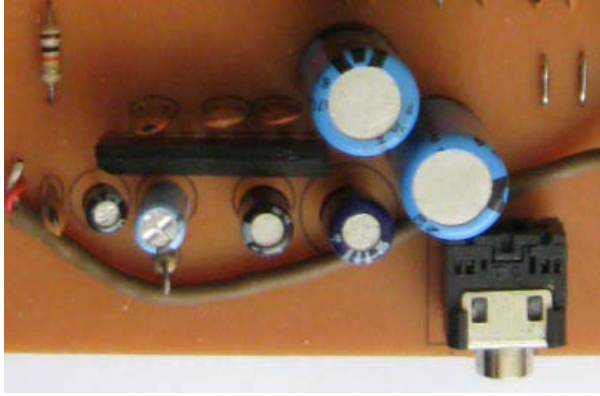
işte bu aradığımız 2. IF frekansımız. Yani yerel osilatör ile 2. IF frekansının toplamı bizim 1. IF frekansımız oluyor,

$$10.7+27=37.7\text{MHz}$$

İyi de madem 37MHz tunerin çıkış filtresinin tepesi, neden onu kullanmadın, diyecek olursanız sebebi şu; hurda kutumda 27MHz'lik kristal vardı ve 37MHz'e en yakın değeri verebilecek, kusursuz tek kristaldi ☺ Başka hiçbir sebebi yok.

### Ses Frekans Yükseltici Katı

Bu devreyi yaparken amacım sadece tuneri test etmekti. Bu nedenle de kullanılacak parçaların piyasada olup olmadığına bakmadım. Hurda kutularımda ne varsa ortaya döküp aralarından işime gelenleri seçtim ve devreyi de ona göre hazırladım. Ses yükseltme katı belki de bu konudaki en sıkıntılı kat. AN7112 entegresi bildiğim kadarıyla çok uzun yıllar önce üretimi durdurulmuş ve piyasada bulmanın gerçekten zor olacağını düşündüğüm bir entegre. Eğer bulabilirsanız yine tükenmiş bir entegre olan LA4140 da buraya uyar, iki entegre birbirlerinin karşılıkları. Hiçbiri olmazsa tek çare elinizdeki yükselteç entegresine göre baskı devreyi yeniden düzenlemek.



Devrede yükselteç arkada kaldı. Ses ayarını öne getirmek için en basit yol olarak potansiyometre ile yükselteç katı arasına çift iletkenli ve yalıtımlı bir kablo çektim. Bunun yerine eski bir kulaklıktan sökeceğiniz bir parça stereo kablo da işi görecektir.

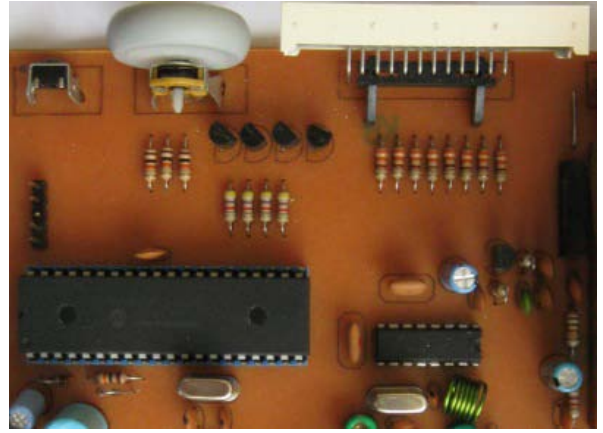
### Kontrol Katı

Kontrol katında bir 16F877 kullanılıyor ama bunu kullanma nedenim sadece işlerimi kolaylaştırmak ve elimin altında olması. Yoksa, bir iki entegre ilavesiyle birlikte bir 16F628 bile buradaki işi fazlasıyla görebilirdi. Sonuçta yapması gereken tunere veri yollamak, LED göstergede bir şeyler yazdırmak ve buton ile rotary anahtardan gelen gelecek sinyalleri yorumlamak. Benim elimde 20MHz'lik olanı vardı, o nedenle – yine elimdeki kristal seçenekleri yüzünden ☺ - 18MHz'lik bir kristal kullandım. Eğer elinizde 4MHz'lik olanlarından varsa programda küçük bir değişiklik ve uygun kristal ile onu da kullanabilirsiniz.

LED gösterge standart türlerden, o nedenle kolayca bulabilirsiniz. Ya da isterseniz kendi elinizdeki göstergeyi buraya uyarlayabilirsiniz. Kritik bir parça değil.

Ayar için kullanılan rotary anahtar eski bir farenin tekeri. Bilgisayarınız varsa ve en az bir yıl kullanmışsanız mutlaka birini bozmuşsunuzdur ☺ Sizde yoksa komşunuzda kesin vardır ☺.

Transistörler biraz garip durdular ama olsun, önemli olan çalışmaları.



Öncelikle ayar işleminden bahsedelim. Radyoyu açtığınızda başlangıç frekansı 45MHz oluyor. Başlangıç frekans adımları ise 50KHz. Dikkat ettiyseniz gösterge üzerinde sadece 4 basamak yer alıyor ve örneğin 100MHz frekansı 50KHz'lik adımlarla gösterebilmek için en az 5 basamağa ihtiyaç var. Bunu burada şu şekilde halettim; en sağdaki basamak 100KHz'lik adımları gösteriyor. Megahertz basamakları ise geri kalan soldaki üçlü oluyor. Frekansın 10KHz'ler basamağı ya 50KHz olacak ya da 0, sonuçta adımlar 50'er Kilohertz. Bu nedenle bu iş için basamak ayırmaya gerek yok, en sağdaki basamağın noktası 50KHz göstergesi olarak kullanılabilir- ki öyle de oldu -. Eğer bu nokta ışıklıysa bilin ki frekans göstergesinde bir de 50KHz var.

Sağdan 2. Basamağın noktası sürekli asyif ve Megahertz basamakları ile Kilohertz basamaklarını ayırıyor.

Devrenin önünde sağda yer alan buton frekans adımlarını 50KHz ya da 1MHz olarak seçmemize yarıyor. Bu butona birinci kez bastığınızda LED göstergenin en soldaki basamağının noktası ışıldar. Bu durumda rotary anahtarı çevirdikçe frekanslar 1'er Megahertz artar ya da azalır. İkinci kez bu butona basıldığında ise bu nokta söner ve adımlar yine 50KHz olur. Bu işlem sayesinde istediğimiz frekansı hızlı bir şekilde bulabiliriz.



1MHz/50KHz  
adım seçimi

MHz/KHz ayırıcı  
nokta. Sabit.

50KHz göstergesi



## Program

Mikrodenetleyicinin programlanması CCSC üzerinde tamamen C dili kullanılarak gerçekleştirildi. Program kodlarını vermeden önce tunerin programlamasıyla ilgili bir iki noktayı vermek istiyorum.

Tuner daha önce de bahsettiğim gibi I<sup>2</sup>C veri yolu kullanarak kontrol ediliyor. PIC16F877 içerisinde I<sup>2</sup>C veri yolu kontrolcüsü donanım

olarak olmasına rağmen ben bunu daha önce hazırladığım I<sup>2</sup>C fonksiyonlarıyla gerçekleştirdim. Bu fonksiyonları diğer PIC türevleriyle de kullanabilirsiniz. Ayrıca donanım yoluyla bu işi gerçekleştirdiğimizde denetleyicideki belli pinleri kullanma zorunluluğu vardır ama fonksiyonlarda istediğiniz pini bu iş için ayarlayabilirsiniz.

Aşağıda tunerin kontrolü için kullanacağımız kaydediciler görülmüyor.

BYTE	BITS								
	7 <sup>(1)</sup> MSB	6	5	4	3	2	1	0 LSB	A <sup>(2)</sup>
Address byte	1	1	0	0	0	MA1	MA2	R/W <sup>(3)</sup>	A
Program divider byte 1	0	n14	n13	n12	n11	n10	n9	n8	A
Program divider byte 2	n7	n6	n5	n4	n3	n2	n1	n0	A
Control information byte1	1	CP	T2	T1	T0	RSA	RSB	0	A
Control information byte 2	X	X	X		BS3	BS2	BS1	BS0	A

Burada Address Byte, tunerle iletişimi başlatmak için gerekli olan ve tunere özgü bir adrestir. R/W biti gönderilecek verinin yazma amaçlı mı yoksa okuma amaçlı mı olduğunu seçmeye yarar. MA bitleri ise çoklu tuner kullanımında işe yarayan bitlerdir ve şu anda bizimle bir işleri yok. Burada bizi en çok meşgul edecek kaydediciler nX ile gösterilen bitlere sahip programlanabilir bölücü kaydedicileridir. Frekans bilgilerini buraya yazacağız. Diğerlerinin bir kısmını da çeşitli ayarlar için kullanacağız. Ama önce frekans hesabına bakalım.

### Frekans Bölücü Hesaplama

İlk sayfada köşede yaptığım küçük açıklamayı okuduysanız burayı anlamak daha kolay olacak. Elimizde 50KHz değerinde bir referans sinyali var. Biz bununla, örneğin 100MHz'i dinlemek istiyoruz. Süperheterodin açıklamasından da hatırlarsanız karıştırıcıya iki sinyal giriyordu ve bunların farkı alınıyordu. 1. IF frekansımız 37.7MHz olduğuna göre 100MHz ile karışacak sinyalden bu değeri elde edebilmek için ya 100MHz'den 37.7MHz yukarı ya da aşağı bir yerel osilatör sinyaline ihtiyacımız olacak. Tuner için bu değer 37.7MHz yukarı olmalı. Yani:

$$100+37.7=137.7\text{MHz}$$

Bu değer 50KHz'in kaç katı olduğunu bulalım.

$$137.7\text{MHz} = 137700\text{KHz}$$
$$137700/50=2754$$

Buradan görüldüğü gibi 100MHz'i dinleyebilmek için tunerimizin programlanabilir bölücüsüne 2754 değerini yüklemek gerek.

Tunerimizin 3 farklı referans frekansını desteklediğinden bahsetmiştik. Bunları RS bitleri ile seçebiliriz.

RSA	RSB	REFERENCE DIVIDER	STEP SIZE (kHz)
X	0	640	50.00
0	1	1024	31.25
1	1	512	62.50

Görüldüğü gibi RSB bitini 0 yapmamız yeterli, bu durumda RSA bitinin bir önemi kalmıyor ve referans frekansımız 50KHz oluyor.

Tunerimizde 3 bant kullanılıyor, bunlar birbiriyle bitişik 3 frekans alanını kapsıyor. Frekans ayarı yaparken de buna dikkat etmek gerek, örneğin 45MHz'i dinliyorsak Alçak Bandı 500MHz'i dinliyorsak Yüksek Bandı seçmeliyiz. Bunu da BS bitleri ile gerçekleştiriyoruz.

BIT								ACTIVE PORT	BAND
X MSB	X	X	X	BS3	BS2	BS1	BS0 LSB		
X	X	X	X	X	0	0	1	BS0	Low band
X	X	X	X	X	0	1	0	BS1	Mid band
X	X	X	X	X	1	0	0	BS2	High band

Tunerde ayrıca faz karşılaştırıcının yük pompası akımını kontrol etme vs. gibi başka ayarlar ve okuma yapabileceğimiz bir durum kaydedicisi var ancak bu uygulamamızda bunları kullanmadık.

Buradaki bilgileri tuner kontrolünün nasıl yapıldığına dair bir fikir verebilmek için yazdım, bu nedenle de detaylara inmiyorum. Hem program gerekli her şeyi yapıyor ve özellikle yeni başlayanların kafalarını şimdilik bunlarla karıştırmaya gerek yok.

# Kaynak Kodlar

## *Ana Program*

// Bu program Murat KAYA tarafından hazırlanmıştır.

```
#include "D:\Elektronik\Projelerim\45-860 Geniş Bant\C\45-860GB.h"
#include "D:\Elektronik\Projelerim\45-860 Geniş Bant\C\i2c.c" // I2C veri yolu kontrol fonksiyonlarını içeren kaynak dosyası.
#include "D:\Elektronik\Projelerim\45-860 Geniş Bant\C\led.c" // 4x7 parçalı gösterge kullanımıyla ilgili fonksiyonları içeren kaynak dosyası.
```

```
#define TUNER_ADR 0xC2 // Tuner I2C adresi
```

```
int8 band[3]={0x01,0x02,0x04}; // Tuner band anahtarları
int32 frekans=4500; // Başlangıç değeri 45MHz
```

```
struct tuner { // Tuner kaydedicilerini içeren yapı
    int16 bolen;
    char kontrol1;
    char kontrol2;}tuner_kontrol;
```

```
void i2c_word_gonder(int16 gidecek) // I2C üzerinden 16 bit gönderen fonksiyon
{
    int8 parca1,parca2;
    parca1=gidecek&0x00FF;
    parca2=(gidecek>>8)&0x00FF;
    i2c_byte_gonder(parca2);
    i2c_byte_gonder(parca1);
}
```

```
void bolen_gonder(void)
{
    i2c_word_gonder(tuner_kontrol.bolen);
}
```

```
void kontrol_gonder(void)
{
    i2c_byte_gonder(tuner_kontrol.kontrol1);
    i2c_byte_gonder(tuner_kontrol.kontrol2);
}
```

```
void i2c_tuner_gonder(int1 anahtar)
{
    if (frekans>=43000) tuner_kontrol.kontrol2=band[2]; // 430MHz'den yüksekse Yüksek Bant
    else if (frekans>=14000) tuner_kontrol.kontrol2=band[1]; // 140MHz'den yüksekse Orta Bant
    else tuner_kontrol.kontrol2=band[0]; // hiçbiri değilse Alçak Bant
    i2c_basla();
    i2c_byte_gonder(TUNER_ADR);
    switch(anahtar)
    {
        case 0:{bolen_gonder();kontrol_gonder();}break;
        case 1:{kontrol_gonder();bolen_gonder();}break;
    }
    i2c_dur();
}
```

```
void frekans_arttir(void) // 50KHz frekans arttır.
{
    if (frekans<86000)
    {
        frekans += 5;
        tuner_kontrol.bolen++;
        i2c_tuner_gonder(0x00);
    }
}
```

```
void frekans_azalt(void) // 50KHz frekans azalt.
{
    if(frekans>4500)
    {
        frekans -= 5;
        tuner_kontrol.bolen--;
        i2c_tuner_gonder(0x01);
    }
}
```

```
void frekans_mhz_arttir(void) // 1MHz frekans arttır.
{
    if (frekans<85900)
```



```

    {
        frekans+=100;
        tuner_kontrol.bolen+=0x14;
        i2c_tuner_gonder(0x00);
    }
}

void frekans_mhz_azalt(void)          // 1MHz frekans azalt.
{
    if(frekans>4600)
    {
        frekans-=100;
        tuner_kontrol.bolen-=0x14;
        i2c_tuner_gonder(0x01);
    }
}

void main()
{
    BOOLEAN anahtar=TRUE;
    int16 sayac=0;
    int8 durum=0, rotary_anahtar[3], yedek;

    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_2);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    set_tris_a(0x00);
    set_tris_b(0x07);
    set_tris_c(0x00);
    set_tris_d(0x00);
    set_tris_e(0x00);
    disable_interrupts(GLOBAL);

    tuner_kontrol.bolen=0x0676;    // DİKKAT! BU NOKTA ÖNEMLİ... Kullanılacak 1. IF frekansına göre burası ayarlanacak.
    tuner_kontrol.kontrol=0xC8;
    i2c_tuner_gonder(0x00);
    rotary_anahtar[0]=0x00;
    rotary_anahtar[1]=0x00;
    rotary_anahtar[2]=0x00;
    donustur(frekans);
    while(TRUE)
    {
        // Bundan sonraki kısım rotary anahtardan gelen sinyalleri değerlendiriyor.
        yedek=input_b() & 0x03;
        if(rotary_anahtar[0]!=yedek)
        {
            rotary_anahtar[2]=rotary_anahtar[1];
            rotary_anahtar[1]=rotary_anahtar[0];
            rotary_anahtar[0]=yedek;
            if(((rotary_anahtar[2]==0x00)&&(rotary_anahtar[0]==0x03))||((rotary_anahtar[2]==0x03)&&(rotary_anahtar[0]==0x00)))
durum=(rotary_anahtar[2]^rotary_anahtar[1]);
            switch(durum)
            {
                case 1: {if(mhz_artma) frekans_mhz_arttir(); else frekans_arttir();} break;
                case 2: {if(mhz_artma) frekans_mhz_azalt(); else frekans_azalt();} break;
            }
            durum=0;
            donustur(frekans);
        }
    }

    if(input(PIN_B2)==0 && anahtar) // Burası da butondan gelen sinyalleri değerlendiriyor.
    {
        while(input(PIN_B2)==0 && sayac<10000) sayac++;
        if(sayac==10000) mhz_artma=!mhz_artma;
        donustur(frekans);
        sayac=0;
        anahtar=FALSE;
    }
    if(input(PIN_B2)==1 && !anahtar) anahtar=TRUE;
    led_goster();
}
}

```

## ***I<sup>2</sup>C Fonksiyonları***

```
void i2c_bekle(void) {delay_us(20);}
```

```
void i2c_basla(void)
{
    i2c_bekle();
    output_low(SCL);
    i2c_bekle();
    output_high(SDA);
    i2c_bekle();
    output_high(SCL);
    i2c_bekle();
    output_low(SDA);
    i2c_bekle();
    output_low(SCL);
    i2c_bekle();
}
```

```
void i2c_dur(void)
{
    i2c_bekle();
    output_low(SCL);
    i2c_bekle();
    output_low(SDA);
    i2c_bekle();
    output_high(SCL);
    i2c_bekle();
    output_high(SDA);
    i2c_bekle();
    output_low(SCL);
}
```

```
void i2c_bit_gonder(BOOLEAN bit)
{
    i2c_bekle();
    output_bit(SDA,bit);
    i2c_bekle();
    output_high(SCL);
    i2c_bekle();
    output_low(SCL);
}
```

```
BOOLEAN i2c_bit_oku (void)
{
    BOOLEAN a;
    i2c_bekle();
    set_tris_a(0x10);
    output_high(SCL);
    i2c_bekle();
    a=input(SDA);
    output_low(SCL);
    i2c_bekle();
    set_tris_a(0x00);
    return a;
}
```

```
BOOLEAN i2c_byte_gonder(char giden)
{
    char sayac;
    BOOLEAN onay;
    for(sayac=0;sayac<8;sayac++)i2c_bit_gonder(shift_left(&giden,1,0));
    onay=i2c_bit_oku();
    return onay;
}
```

```
char i2c_byte_oku(BOOLEAN onay)
{
    char gidecek=0,sayac;
    for(sayac=0;sayac<8;sayac++)shift_left(&gidecek,1,i2c_bit_oku());
    i2c_bit_gonder(onay);
    return gidecek;
}
```

```
BOOLEAN i2c_kontrol(char cih_kodu)
{
    BOOLEAN onay;
```

```
i2c_basla();
onay=i2c_byte_gonder(cih_kodu);
i2c_dur();
return onay;
}
```

## ***LED Gösterge Fonksiyonları***

// 7 parçalı LED gösterge için veri görüntüleme fonksiyonları

int8 desen[10]={0x14, 0xD7, 0x4C, 0x45, 0x87, 0x25, 0x24, 0x57, 0x04, 0x05}; // 7 parçalı göstergede rakamları vermek için gerekli çıkış bit desenleri.

```
int8 basamaklar[5];
BOOLEAN mhz_artma=FALSE;
```

```
void led_goster(void)
```

```
{
    static int8 sayac=4;

    output_d(0xFF);
    output_c((0x10<<(sayac-1))^0xF0);
    output_d(basamaklar[sayac]);
    if(sayac!=0x01) sayac--;
    else sayac=4;
}
```

```
void donustur(int32 veri)
```

```
{
    int8 sayac;

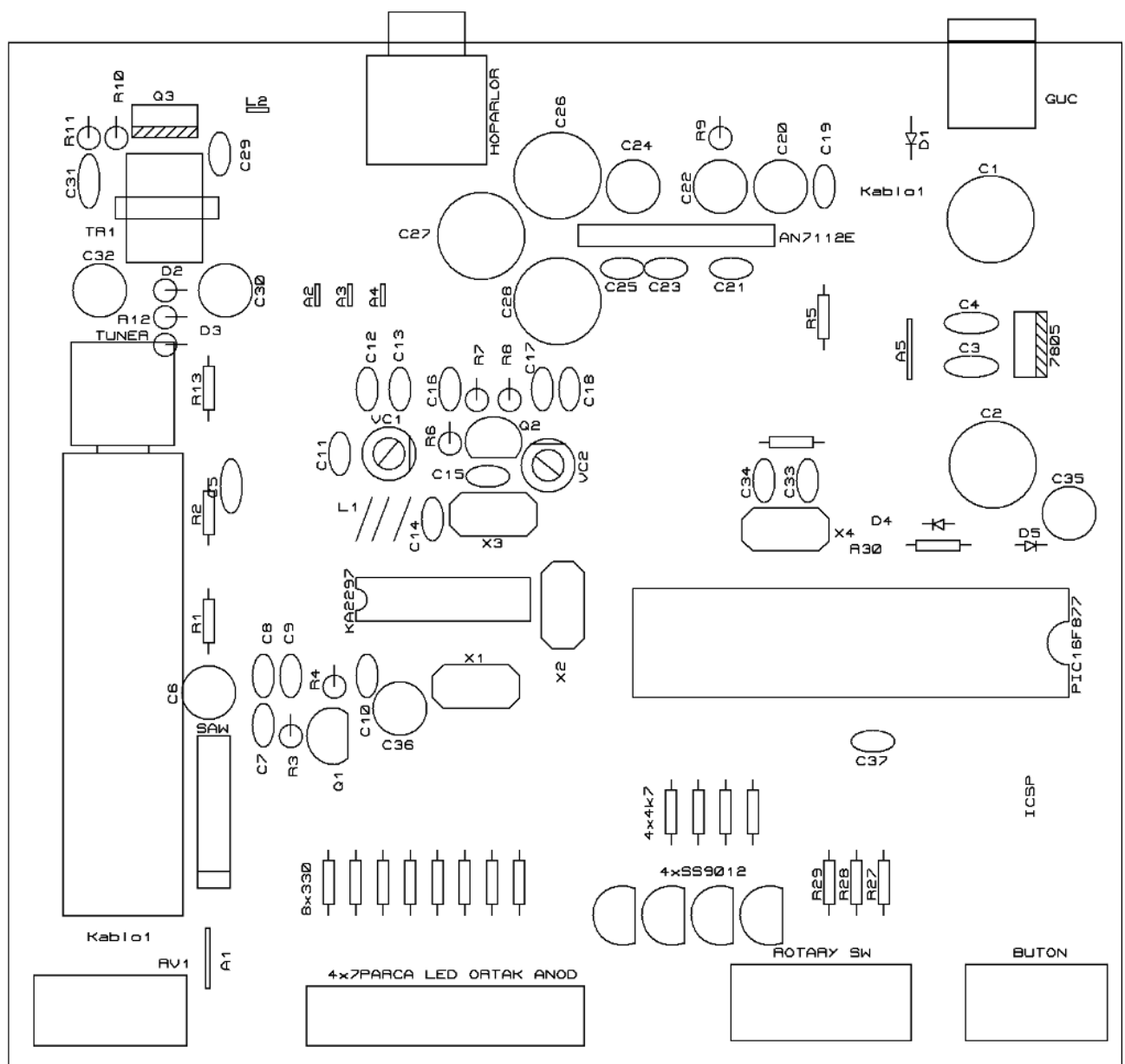
    if(veri<100000)
    {
        for(sayac=0;sayac<5;sayac++)
        {
            basamaklar[sayac]=desen[veri%10];
            veri/=10;
        }
        sayac=4;
        while((basamaklar[sayac]==0x14)&& sayac>1)
        {
            basamaklar[sayac]=0xFF;
            sayac--;
        }
        basamaklar[2]=basamaklar[2] & 0xFB;
        if(basamaklar[0]==0x25) basamaklar[1]=basamaklar[1] & 0xFB;else basamaklar[1]=basamaklar[1] | 0x04;
        if(mhz_artma)basamaklar[4]=basamaklar[4] & 0xFB;
    }
}
```

## Baskı Devre

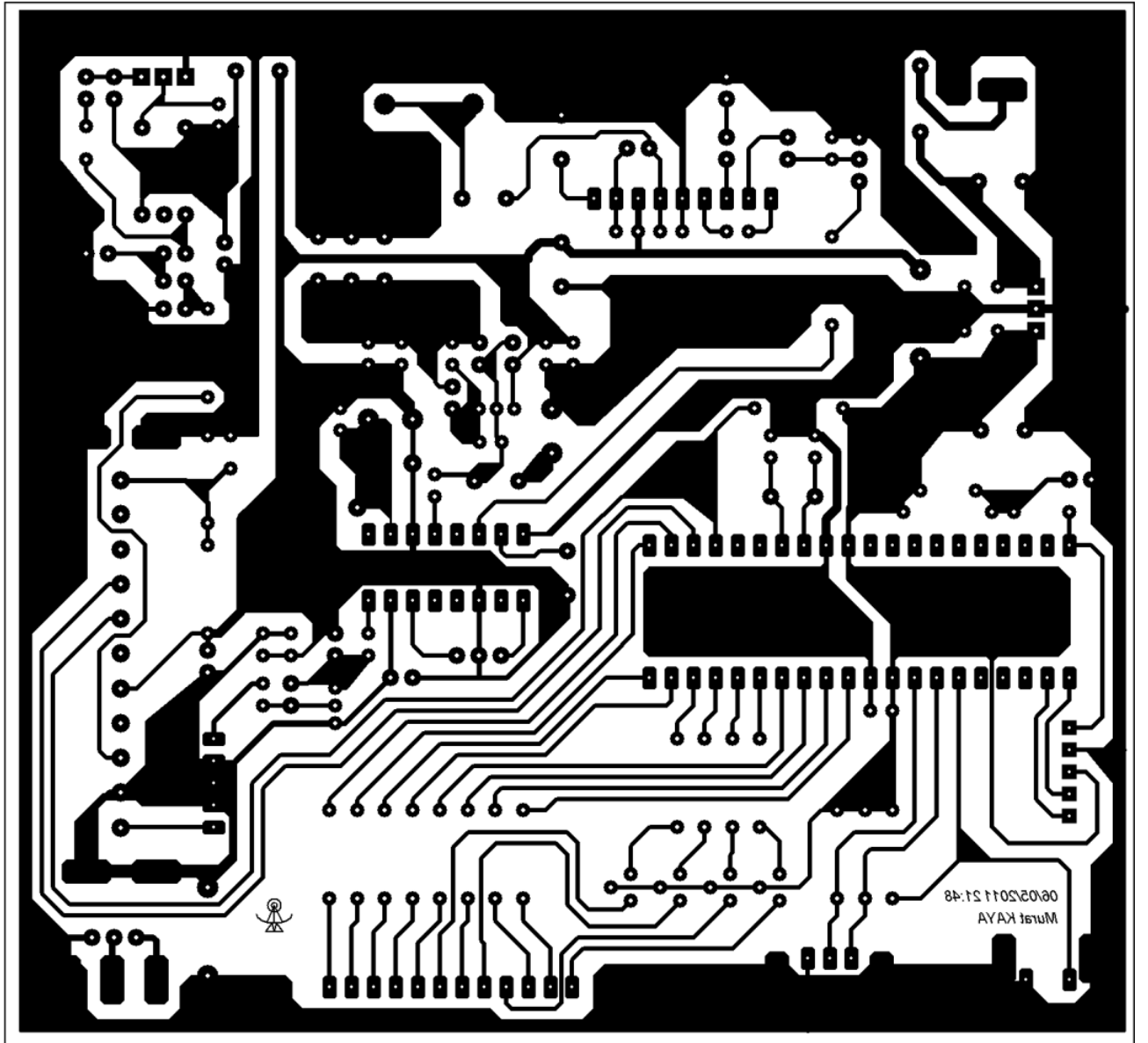
Bu devreyi yapacak kişilerin baskı devre konusunda az veya çok deneyim sahibi oldukları varsayılmıştır. Bu nedenle burada baskı devre yapım teknikleri hakkında bilgi vermek düşünülmemiştir.

Her ne kadar baskı devre şekillerini burada vermiş olsam da esas olarak orijinallerini kullanmanızı tavsiye ederim.

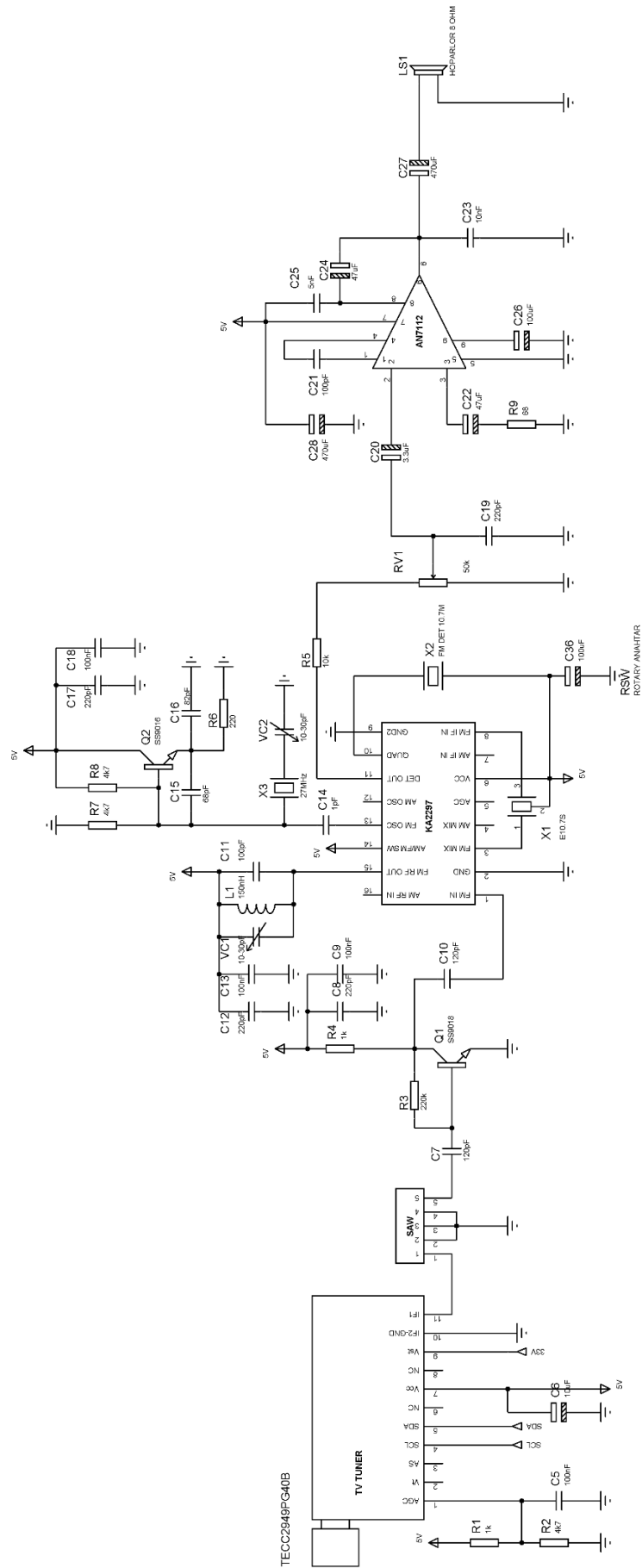
*Üst İpek*



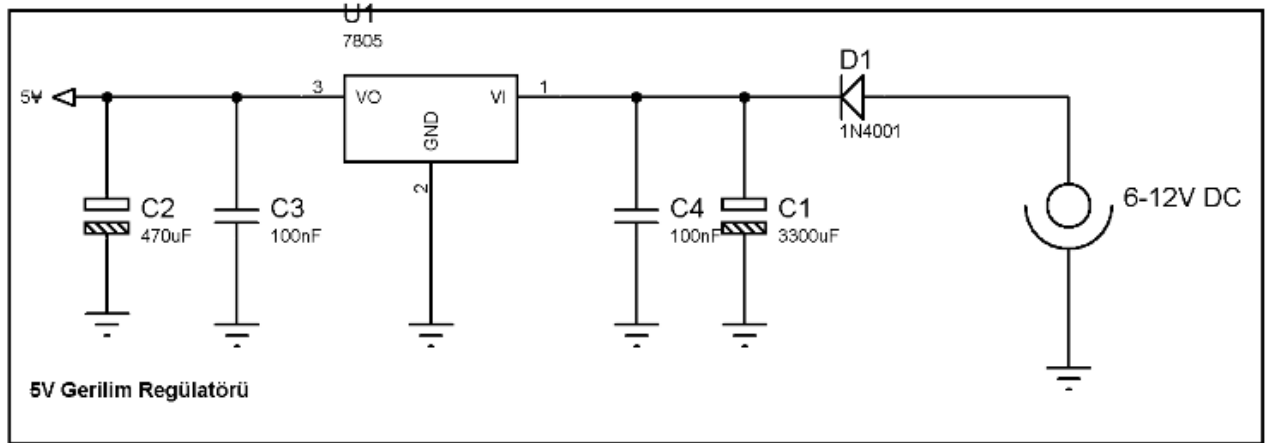




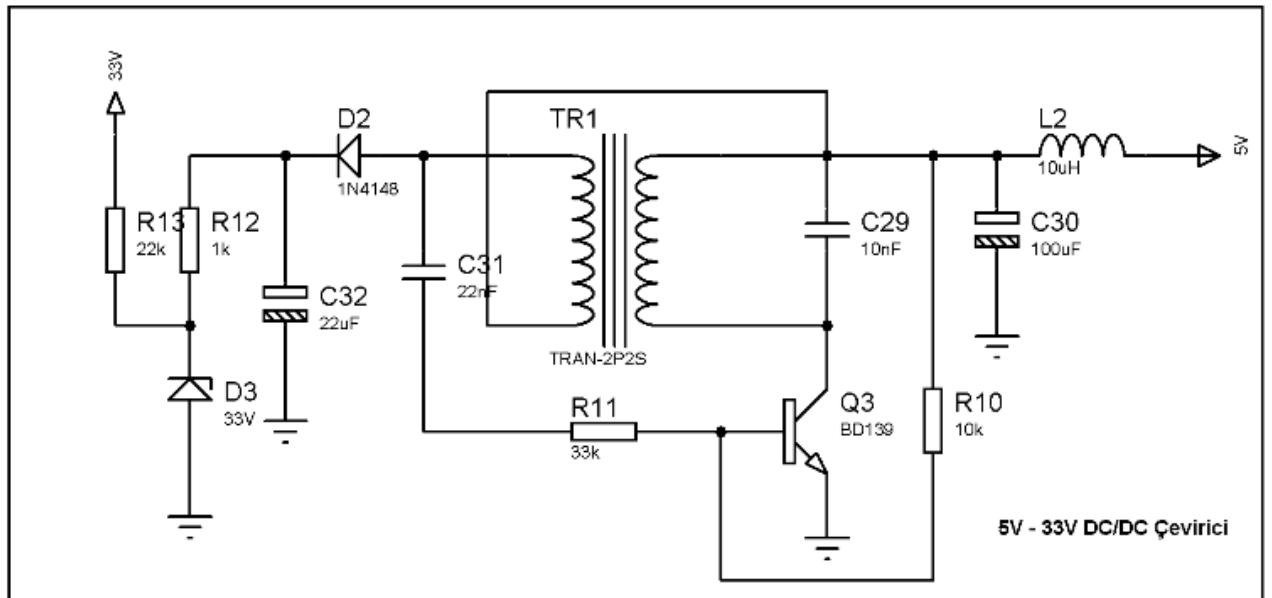
### *Alıcı Kısım*



## Güç Katı



## 33V DC/DC Çevirici



# Sayısal Kontrol Katı

