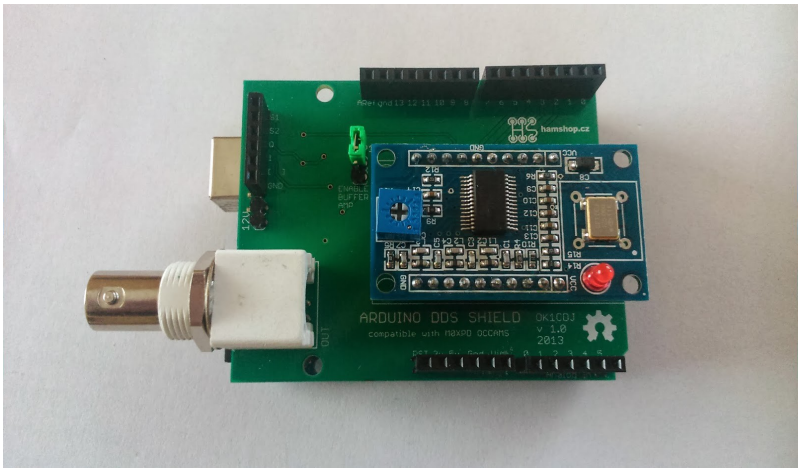
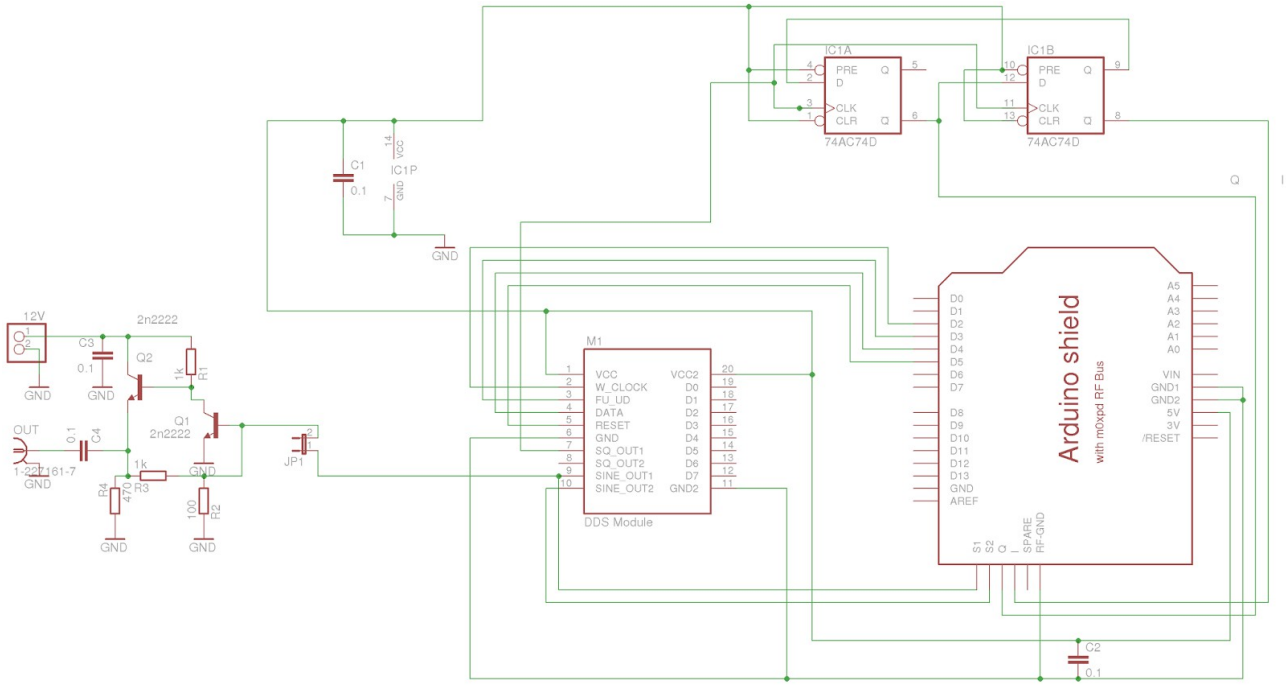
DDS Shield para un Arduino UNO 12 de enero de 2014 Ondra OK1CDJ

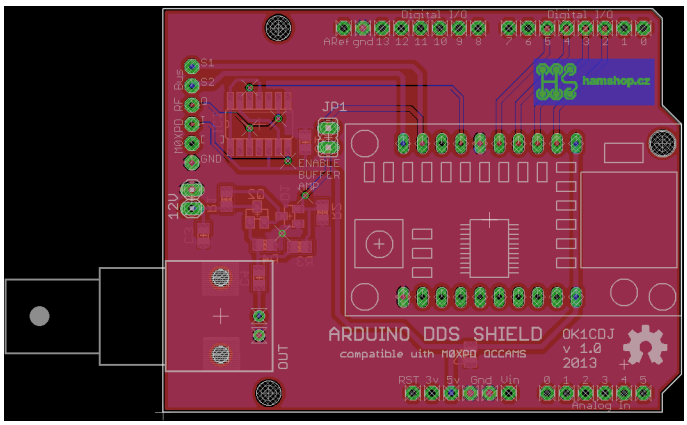


Los módulos DDS con el AD9850 se puede conseguir relativamente baratos. Con uno de ellos se ha construido un "shield" para el Arduino UNO. Además del módulo DDS se ha instalado un amplificador separador capaz de suministrar 2-3 V pico-pico sobre una carga de 50 Ω y un divisor con un 74AC74 que permite obtener salidas I/Q necesarias para un SDR simple como el Softrock o similar.
El "shield" va enchufado directamente al Arduino UNO y se puede usar como generador de RF, como OFV o como baliza para QRSS o WSPR. El "shield" es compatible con el proyecto OCCAM de M0XPD (https://sites.google.com/site/occamsmicrocontroller/) recientemente publicado en el boletín SPRAT del Club G-QRP.

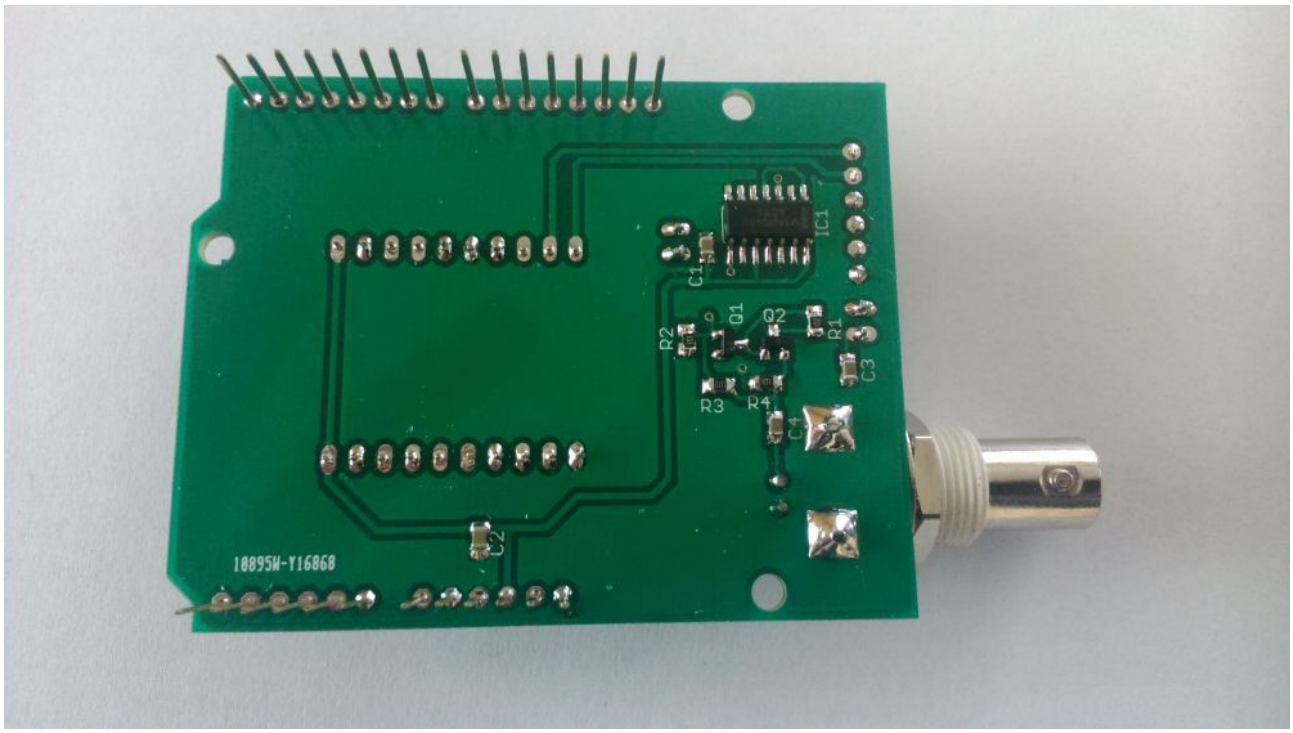
El montaje es sencillo. Instale primero los componentes SMD por la cara inferior del circuito impreso. Monte después los conectores y luego suelde el módulo en su posición lo más pegado a la placa de circuito impreso como sea posible. No es posible usar conectores para el módulo DDS porque impediría por su altura la inserción de módulos (shield) adicionales encima de este.



Esquema del circuito



Circuito impreso



Vista de la cara inferior del circuito impreso

Lista de componentes:

Componente	Valor	Componente	Valor
C1	100 nF	Q1	2N2222
C2	100 nF	Q2	2N2222
C3	100 nF	R1	1 kΩ
C4	100 nF	R2	100 Ω
IC1	74AC74D	R3	1 kΩ
M1	Módulo DDS	R4	470 Ω
OUT	conector BNC		

A continuación se incluye un ejemplo de código a usar para inicializar el módulo DDS y generar una frecuencia de 10 MHz. En Internet podrá obtener programas similares.

Código de ejemplo para Arduino:

```
/*
 * A simple single freq AD9850 Arduino test script
 * Original AD9851 DDS sketch by Andrew Smallbone at www.rocketnumbernine.com
 * Modified for testing the inexpensive AD9850 DDS modules
 * 9850 datasheet at http://www.analog.com/static/imported-
files/data_sheets/AD9850.pdf
 * Use freely
 */

#define W_CLK 2 // Pin 2 - connect to AD9850 module word load clock pin (CLK)
#define FQ_UD 3 // Pin 3 - connect to freq update pin (FQ)
#define DATA 4  // Pin 4 - connect to serial data load pin (DATA)
#define RESET 5 // Pin 5 - connect to reset pin (RST).

#define pulseHigh(pin) {digitalWrite(pin, HIGH); digitalWrite(pin, LOW); }

// transfers a byte, a bit at a time, LSB first to the 9850 via serial DATA line
void tfr_byte(byte data)
{
 for (int i=0; i<8; i++, data>>=1) {
  digitalWrite(DATA, data & 0x01);
  pulseHigh(W_CLK);   //after each bit sent, CLK is pulsed high
 }
}

// frequency calc from datasheet page 8 = <sys clock> * <frequency tuning
//word>/2^32
void sendFrequency(double frequency) {
 int32_t freq = frequency * 4294967295/125000000;  // note 125 MHz clock on 9850
 for (int b=0; b<4; b++, freq>>=8) {
  tfr_byte(freq & 0xFF);
 }
 tfr_byte(0x000);   // Final control byte, all 0 for 9850 chip
 pulseHigh(FQ_UD);  // Done!  Should see output
}

void setup() {
 // configure arduino data pins for output
 pinMode(FQ_UD, OUTPUT);
 pinMode(W_CLK, OUTPUT);
 pinMode(DATA, OUTPUT);
 pinMode(RESET, OUTPUT);

 pulseHigh(RESET);
 pulseHigh(W_CLK);
 pulseHigh(FQ_UD);  // this pulse enables serial mode - Datasheet page 12 figure
//10
}

void loop() {
 sendFrequency(10.e6);  // freq 10.00000 MHz
 while(1);
}
```

(Este kit se puede comprar en Hamshop.cz)