```
% ascii_test_recv.m
%   Tests the complete receiver using an ASCII message.
% Set up standard parameters as you did before
%% System parameters
param = system_param;

% Ascii message to send (feel free to use your own message here!)
ascii_text = 'Stand by .... Stand by .... Stand by .... Four score and seven years ago our fathers
brought forth on this continent a new nation, conceived in Liberty, and dedicated to the proposition
that all men are created equal. Now we are engaged in a great civil war, testing whether that
nation, or any nation, so conceived and so dedicated, can long endure.';

% Convert the ascii message to 4PSK symbols
frm = ascii_to_symb_frame(2, param.frame.sync, param.frame.N, ascii_text);
symb = frm(:);
as = 1;
at = 1;
Ns = 192;
n = 1:Ns;
% Generate signal from symbols
[s s_debug] = make_signal_4psk(param.fs, param.f0, param.ft, param.cps, param.h_ps, as, at, symb);
Nb = floor(length(s)/Ns);
% Make signal into blocks
sb = reshape(s(1:Ns*Nb), Ns, Nb);

% Intialize state of frame synchronization code
frame_state = [];

% Initialize DDC and STR
ddc_state = ddc_init(param.ddc);
str_state = str_init(param.str);

% Process the waveform s as before ...
for ii=1:Nb,
  % Get a single block
  x = sb(:, ii);
  % Digital down converter
  [Ib Qb ddc_state ddc_debug] = ddc(x, ddc_state);
  % Symbol timing recovery
  [si str_state] = str(Ib, Qb, str_state);
  % Sample at optimal points
  si1 = find(si)
  % Decision block
  symb_out = decision(Ib(si1), Qb(si1));
  % Result should now be a vector of symbols (values 0-3) estimated from
  % the optimal sample points for the current block.  I assume here this vector is called
  % "symb_out"
```

```
  % Recover frames.  This code only prints something when a complete frame is ready.
  [frame_out frame_state] = frame(symb_out, param.frame, frame_state);
  if ~isempty(frame_out),
    fprintf('%s', symb_to_ascii(2, frame_out));
  end
  pause;
end
```