

```

#include <SPFD5408_Adafruit_TFTLCD.h>

#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4

// Agar warna mudah dimengerti (Human Readable color):
#define TFT_BLACK 0x0000
#define TFT_BLUE 0x001F
#define TFT_RED 0xF800
#define TFT_GREEN 0x07E0
#define TFT_CYAN 0x07FF
#define TFT_MAGENTA 0xF81F
#define TFT_YELLOW 0xFFE0
#define TFT_WHITE 0xFFFF
#define TFT_GREY 0x5AEB

Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

float sx = 0, sy = 1, mx = 1, my = 0, hx = -1, hy = 0; // Saved H, M, S x & y multipliers
float sdeg = 0, mdeg = 0, hdeg = 0;
uint16_t osx = 120, osy = 120, omx = 120, omy = 120, ohx = 120, ohy = 120; // Saved H, M, S x &
y coords
int16_t x0 = 0, x1 = 0, yy0 = 0, yy1 = 0, x00 = 0, yy00 = 0;
uint32_t targetTime = 0; // for next 1 second timeout

uint16_t xpos; // x posisi jam
uint8_t conv2d(const char* p) {
  uint8_t v = 0;
  if ('0' <= *p && *p <= '9')
    v = *p - '0';
  return 10 * v + *++p - '0';
}
uint8_t hh = conv2d(__TIME__), mm = conv2d(__TIME__ + 3), ss = conv2d(__TIME__ + 6); //
mengambil data waktu dari jam Compile-Upload
boolean initial = 1;
char d;
void setup(void) {
  tft.reset(); //perlu saat menggunakan lib.SPFD5408_Adafruit_TFTLCD.h
  tft.begin(0x9341); //perlu address ini saat menggunakan lib.SPFD5408_Adafruit_TFTLCD.h
  tft.setRotation(3);

  tft.setTextColor(TFT_WHITE); //warna text
  tft.fillScreen(TFT_BLACK); //warna latar

  // Draw clock face
  xpos = tft.width() / 2; // mencari titik koordinat tengah LCD
  tft.drawCircle(xpos, 120, 125, TFT_YELLOW);
  tft.fillCircle(xpos, 120, 118, TFT_BLUE); //warna lingkaran luar
  tft.fillCircle(xpos, 120, 110, TFT_BLACK); //warna jam bagian dalam

```

```

for (int a=95; a<104; a++){
tft.drawCircle(xpos, 120, a, TFT_WHITE);} //warna lingkaran luar II

// Draw 12 lines
for (int i = 0; i < 360; i += 30) {
  sx = cos((i - 90) * 0.0174532925);
  sy = sin((i - 90) * 0.0174532925);
  x0 = sx * 114 + xpos;
  yy0 = sy * 114 + 120;
  x1 = sx * 100 + xpos;
  yy1 = sy * 100 + 120;

  tft.drawLine(x0, yy0, x1, yy1, TFT_YELLOW);//garis penanda angka jam
}

// Draw 60 dots
for (int i = 0; i < 360; i += 6) {
  sx = cos((i - 90) * 0.0174532925);
  sy = sin((i - 90) * 0.0174532925);
  x0 = sx * 102 + xpos;
  yy0 = sy * 102 + 120;
  x00 = sx * 92 + xpos;
  yy00 = sy * 92 + 120;
  // Draw minute markers
  tft.drawPixel(x0, yy0, TFT_GREEN); //titik penanda menit
  tft.drawLine(x0, yy0, x00, yy00, TFT_BLACK);//garis penanda menit
  tft.drawLine(x0+1, yy0+1, x00+1, yy00+1, TFT_BLACK);//garis penanda menit

  // Draw main quadrant dots
  if (i == 0 || i == 180) tft.fillCircle(x0, yy0, 2, TFT_WHITE); //penanda 12 dan 6
  if (i == 90 || i == 270) tft.fillCircle(x0, yy0, 2, TFT_WHITE); //penanda 3 dan 9
}

tft.fillCircle(xpos, 121, 3, TFT_WHITE);
targetTime = millis() + 1000;
}

void loop() {
  if (targetTime < millis()) {
    targetTime = millis() + 1000;
    ss++; // Advance second
    if (ss == 60) {
      ss = 0;
      mm++; // Advance minute
      if (mm > 59) {
        mm = 0;
        hh++; // Advance hour
        if (hh > 23) {
          hh = 0;
        }
      }
    }
  }
}

```

```

    }
}

// Pre-compute hand degrees, x & y coords for a fast screen update
sdeg = ss * 6;           // 0-59 -> 0-354
mdeg = mm * 6 + sdeg * 0.01666667; // 0-59 -> 0-360 - includes seconds, but these increments
are not used
hdeg = hh * 30 + mdeg * 0.08333333; // 0-11 -> 0-360 - includes minutes and seconds, but these
increments are not used
hx = cos((hdeg - 90) * 0.0174532925);
hy = sin((hdeg - 90) * 0.0174532925);
mx = cos((mdeg - 90) * 0.0174532925);
my = sin((mdeg - 90) * 0.0174532925);
sx = cos((sdeg - 90) * 0.0174532925);
sy = sin((sdeg - 90) * 0.0174532925);

if (ss == 0 || initial) {
    initial = 0;
    // Erase hour and minute hand positions every minute
    tft.drawLine(ohx, ohy, xpos, 121, TFT_BLACK);
    ohx = hx * 62 + xpos + 1;
    ohy = hy * 62 + 121;
    tft.drawLine(omx, omy, xpos, 121, TFT_BLACK);
    omx = mx * 84 + xpos;
    omy = my * 84 + 121;
}

// Redraw new hand positions, hour and minute hands not erased here to avoid flicker
tft.drawLine(osx, osy, xpos, 121, TFT_BLACK);
osx = sx * 90 + xpos + 1;
osy = sy * 90 + 121;
tft.drawLine(osx, osy, xpos, 121, TFT_RED);
tft.drawLine(ohx, ohy, xpos, 121, TFT_CYAN);
tft.drawLine(omx, omy, xpos, 121, TFT_WHITE);
tft.drawLine(osx, osy, xpos, 121, TFT_RED);
tft.fillCircle(xpos, 121, 3, TFT_RED);

tft.setCursor(xpos-30, 45);
tft.setTextSize(1);
tft.print("SMARTPOKER");

// Draw MINI clock face "SECOND"
tft.drawCircle(xpos, 155, 20, TFT_YELLOW);
tft.drawCircle(xpos, 155, 18, TFT_BLUE);
tft.drawCircle(xpos, 155, 17, TFT_CYAN);
tft.drawCircle(xpos, 155, 16, TFT_CYAN);
tft.fillRect(xpos-10, 149, 22, 15, TFT_BLACK); //erase
if(ss<10){tft.setCursor(xpos-10, 149); tft.setTextSize(2);
tft.print('0'); tft.setCursor(xpos+2, 149);}
else{
tft.setCursor(xpos-10, 149);}
tft.setTextSize(2);

```

```

tft.print(ss);

// Draw MINI clock face "Minutes"
tft.drawCircle(xpos+35, 117, 20, TFT_YELLOW);
tft.drawCircle(xpos+35, 117, 18, TFT_BLUE);
tft.drawCircle(xpos+35, 117, 17, TFT_CYAN);
tft.drawCircle(xpos+35, 117, 16, TFT_CYAN);
tft.fillRect(xpos+25, 111,22,15,TFT_BLACK); //erase
if(mm<10){tft.setCursor(xpos+25, 111); tft.setTextSize(2);
tft.print('0'); tft.setCursor(xpos+37, 111);}
else{
tft.setCursor(xpos+25, 111);}
tft.println(mm);

// Draw MINI clock face "Hour"
tft.drawCircle(xpos-35, 117, 20, TFT_YELLOW);
tft.drawCircle(xpos-35, 117, 18, TFT_BLUE);
tft.drawCircle(xpos-35, 117, 17, TFT_CYAN);
tft.drawCircle(xpos-35, 117, 16, TFT_CYAN);
tft.fillRect(xpos-45, 111,22,15,TFT_BLACK); //erase
if(hh<10){tft.setCursor(xpos-45, 111); tft.setTextSize(2);
tft.print('0'); tft.setCursor(xpos-33, 111);}
else{
tft.setCursor(xpos-45, 111);}
tft.setTextSize(2);
tft.print(hh);
//tft.setCursor(xpos-65, 111);
//tft.println(':');

if (hh>=0 && hh<12) d='A'; else {d='P'};
tft.drawRoundRect(xpos-14,72,29,21,5,TFT_CYAN);
tft.fillRect(xpos-11, 75,23,15,TFT_BLACK); //erase
tft.setCursor(xpos-11, 75);
tft.setTextSize(2);
tft.print(d);
tft.println('M');
}
}

```