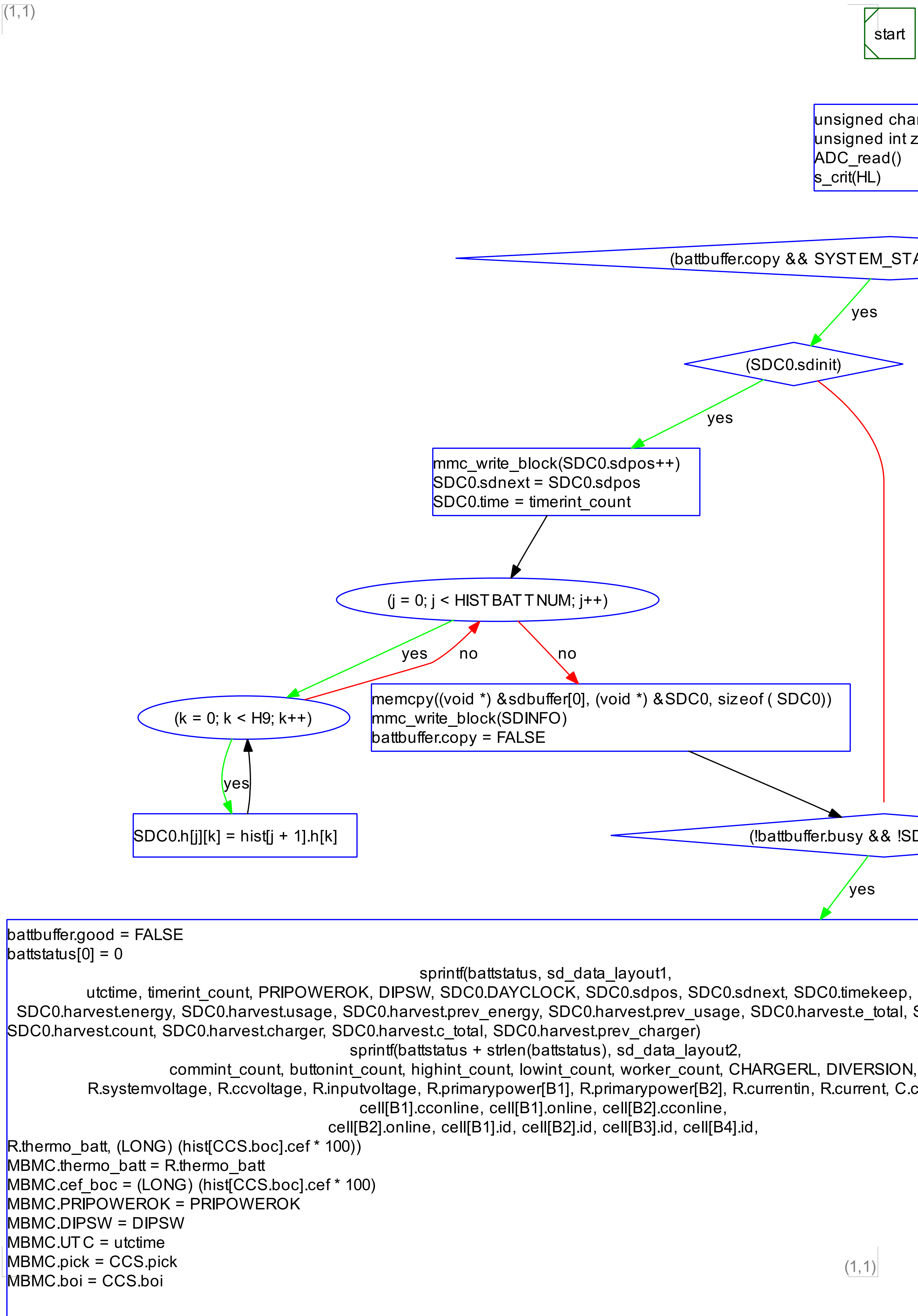
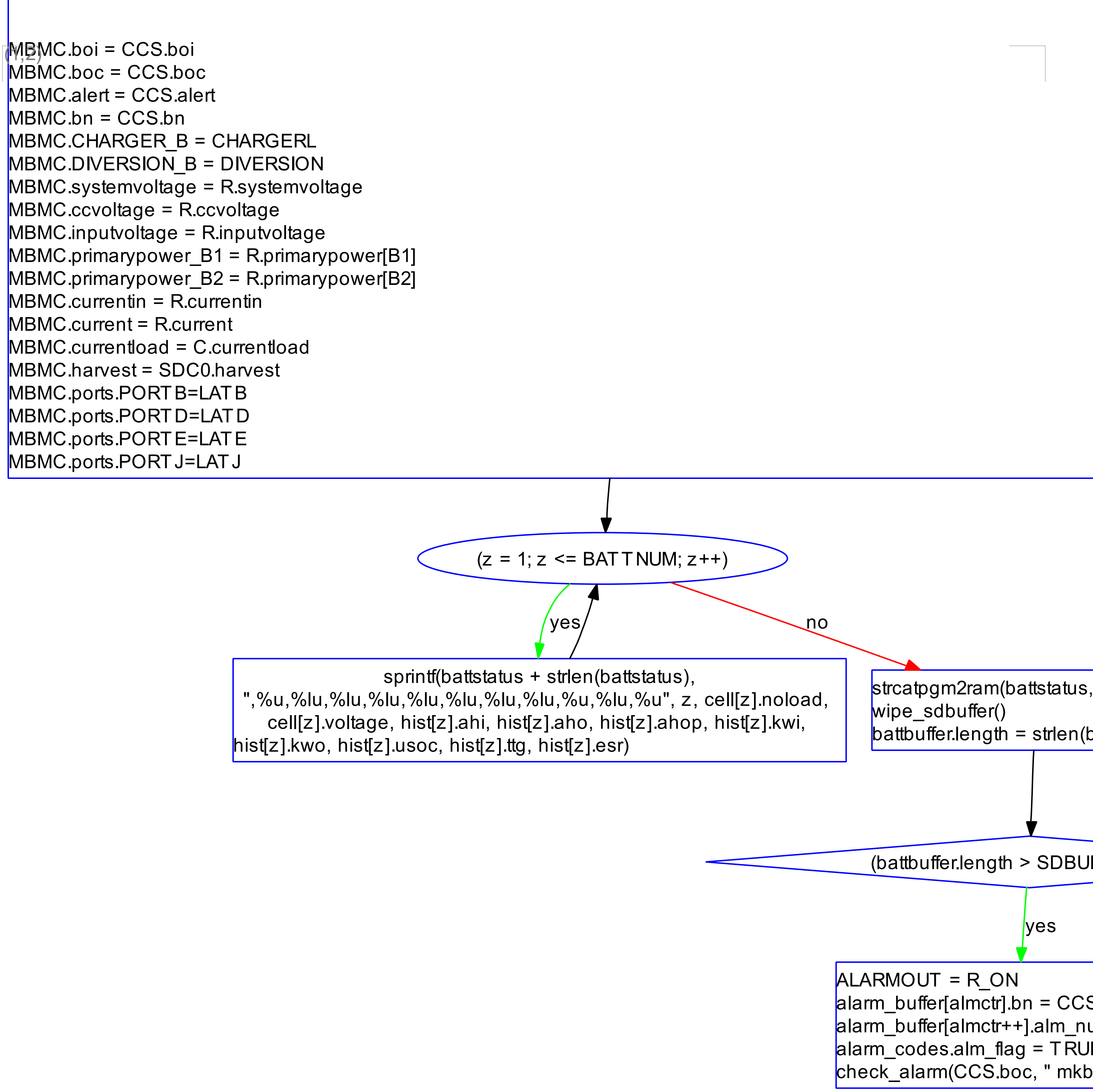


```
unsigned char
unsigned int z
ADC_read()
s_crit(HL)

(battbuffer.copy && SYSTEM_STA

(SDC0.sdinit)

mmc_write_block(SDC0.sdpos++)
SDC0.sdnext = SDC0.sdpos
SDC0.time = timerint_count

(j = 0; j < HISTBATTNUM; j++)

(k = 0; k < H9; k++)

memcpy((void *) &sdbuffer[0], (void *) &SDC0, sizeof ( SDC0))
mmc_write_block(SDINFO)
battbuffer.copy = FALSE

SDC0.h[j][k] = hist[j + 1].h[k]

(!battbuffer.busy && !SD

battbuffer.good = FALSE
battstatus[0] = 0
                                        sprintf(battstatus, sd_data_layout1,
           utctime, timerint_count, PRIPOWEROK, DIPSW, SDC0.DAYCLOCK, SDC0.sdpos, SDC0.sdnext, SDC0.timekeep,
 SDC0.harvest.energy, SDC0.harvest.usage, SDC0.harvest.prev_energy, SDC0.harvest.prev_usage, SDC0.harvest.e_total, S
SDC0.harvest.count, SDC0.harvest.charger, SDC0.harvest.c_total, SDC0.harvest.prev_charger)
                                  sprintf(battstatus + strlen(battstatus), sd_data_layout2,
                  commint_count, buttonint_count, highint_count, lowint_count, worker_count, CHARGERL, DIVERSION,
           R.systemvoltage, R.ccvoltage, R.inputvoltage, R.primarypower[B1], R.primarypower[B2], R.currentin, R.current, C.c
                                   cell[B1].cconline, cell[B1].online, cell[B2].cconline,
                                cell[B2].online, cell[B1].id, cell[B2].id, cell[B3].id, cell[B4].id,
R.thermo_batt, (LONG) (hist[CCS.boc].cef * 100))
MBMC.thermo_batt = R.thermo_batt
MBMC.cef_boc = (LONG) (hist[CCS.boc].cef * 100)
MBMC.PRIPOWEROK = PRIPOWEROK
MBMC.DIPSW = DIPSW
MBMC.UTC = utctime
MBMC.pick = CCS.pick
MBMC.boi = CCS.boi
```



```
MBMC.boi = CCS.boi
MBMC.boc = CCS.boc
MBMC.alert = CCS.alert
MBMC.bn = CCS.bn
MBMC.CHARGER_B = CHARGERL
MBMC.DIVERSION_B = DIVERSION
MBMC.systemvoltage = R.systemvoltage
MBMC.ccvoltage = R.ccvoltage
MBMC.inputvoltage = R.inputvoltage
MBMC.primarypower_B1 = R.primarypower[B1]
MBMC.primarypower_B2 = R.primarypower[B2]
MBMC.currentin = R.currentin
MBMC.current = R.current
MBMC.currentload = C.currentload
MBMC.harvest = SDC0.harvest
MBMC.ports.PORTB=LATB
MBMC.ports.PORTD=LATD
MBMC.ports.PORTE=LATE
MBMC.ports.PORTJ=LATJ
```

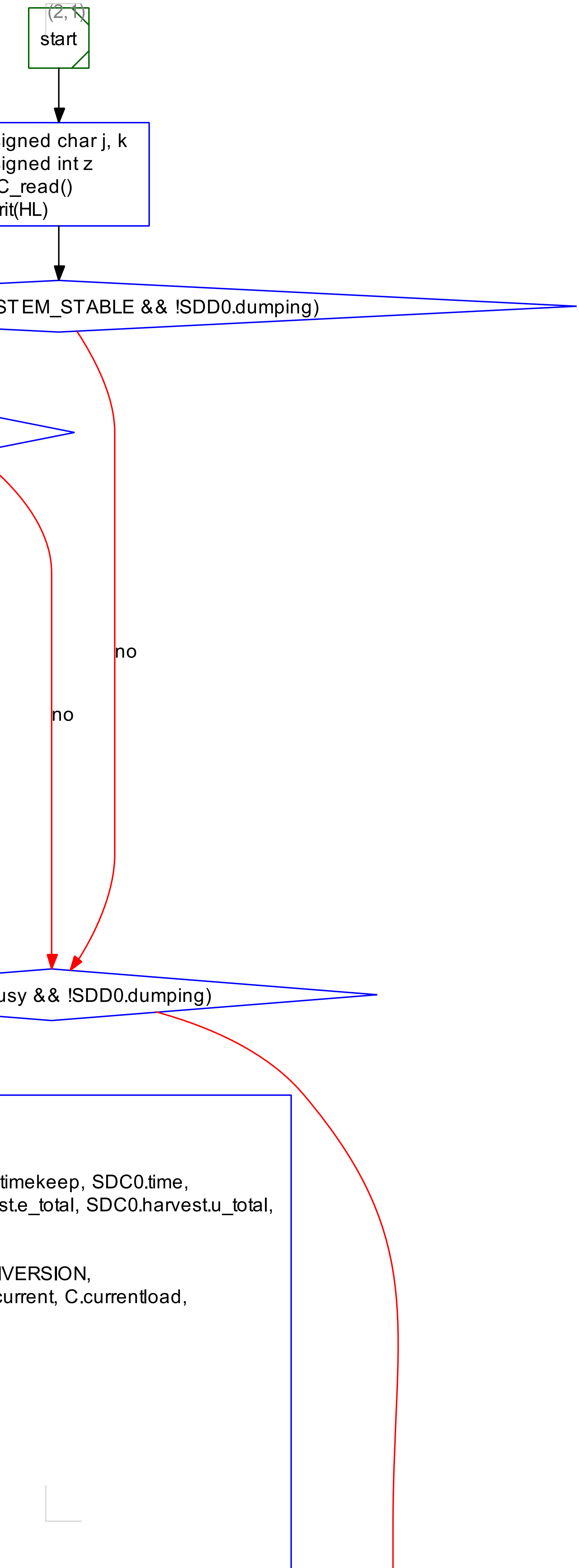
```
(z = 1; z <= BATTNUM; z++)
```

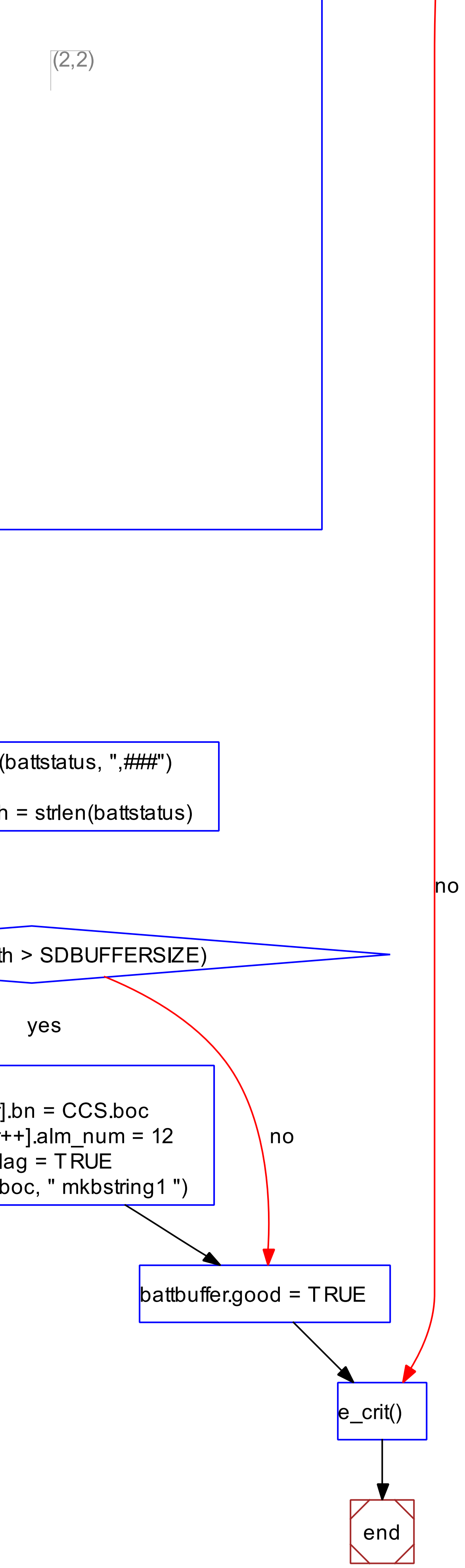
```
sprintf(battstatus + strlen(battstatus),
",%u,%lu,%lu,%lu,%lu,%lu,%lu,%lu,%u,%lu,%u", z, cell[z].noload,
cell[z].voltage, hist[z].ahi, hist[z].aho, hist[z].ahop, hist[z].kwi,
hist[z].kwo, hist[z].usoc, hist[z].ttg, hist[z].esr)
```

```
strcatpgm2ram(battstatus,
wipe_sdbuffer()
battbuffer.length = strlen(b
```

```
(battbuffer.length > SDBU
```

```
ALARMOUT = R_ON
alarm_buffer[almctr].bn = CCS
alarm_buffer[almctr++].alm_nu
alarm_codes.alm_flag = TRU
check_alarm(CCS.boc, " mkb
```


(2, 1)
start
igned char j, k
igned int z
C_read()
rit(HL)
STEM_STABLE && !SDD0.dumping)
no
no
usy && !SDD0.dumping)
timekeep, SDC0.time,
st.e_total, SDC0.harvest.u_total,
VERSION,
current, C.currentload,


(battstatus, ",###")
n = strlen(battstatus)
no
th > SDBUFFERSIZE)
yes
].bn = CCS.boc
++].alm_num = 12
lag = TRUE
boc, " mkbstring1 ")
no
battbuffer.good = TRUE
e_crit()
end