

Push0_15

Using the indirect addressing mode instructions `ld` and `st`, push registers `r0` to `r15` onto a stack buffer starting at SRAM address `0x04FF`. To test your subroutine, begin by adding and then calling subroutine `PreReg0_20`, in the setup section of your code.

X pointer	0x0015
Y pointer	0x0500
Z pointer	0x0000
Cycle Counter	115
Frequency	1.0000 MHz
Stop Watch	115.00 us
SREG	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Registers	
R00	0x15
R01	0x14
R02	0x13
R03	0x12
R04	0x11
R05	0x10
R06	0x0F
R07	0x0E
R08	0x0D
R09	0x0C
R10	0x0B
R11	0x0A
R12	0x09
R13	0x08
R14	0x07
R15	0x06
R16	0x05

```

.INCLUDE <m328pdef.inc>

Setup:
  ldi YH,0x05 ; YH = r29, initialize register
  ldi YL,0x00 ; YL = r28, pre-decrement, first

  rcall PreReg0_20
  rcall SaveReg0_15

done:
  rjmp done

SaveReg0_15:
  clr XH ; r27
  clr XL ; r26
  clr r25 ; starting with r0
  ldi r24,16 ; save 16 registers

save_reg:
  ld r25, X+
  st -Y, r25
  dec r24
  brne save_reg
  ret
    
```

Data	8/16	abc.	Address: 0x4EF	Cols: Auto											
0004EF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000507	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00051F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000537	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 1 Registers r0 to r20 preset to values 0x15 to 0x01.

X pointer	0x0010
Y pointer	0x04F0
Z pointer	0x0000
Cycle Counter	237
Frequency	1.0000 MHz
Stop Watch	237.00 us
SREG	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Registers	
R00	0x15
R01	0x14
R02	0x13
R03	0x12
R04	0x11
R05	0x10
R06	0x0F
R07	0x0E
R08	0x0D
R09	0x0C
R10	0x0B
R11	0x0A
R12	0x09
R13	0x08
R14	0x07
R15	0x06
R16	0x05

```

.INCLUDE <m328pdef.inc>

Setup:
  ldi YH,0x05 ; YH = r29, initialize register
  ldi YL,0x00 ; YL = r28, pre-decrement, first

  rcall PreReg0_20
  rcall SaveReg0_15

done:
  rjmp done

SaveReg0_15:
  clr XH ; r27
  clr XL ; r26
  clr r25 ; starting with r0
  ldi r24,16 ; save 16 registers

save_reg:
  ld r25, X+
  st -Y, r25
  dec r24
  brne save_reg
  ret
    
```

Data	8/16	abc.	Address: 0x4EF	Cols: Auto													
0004EF	00	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
000507	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00051F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000537	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 2 Registers r0 to r20 saved to stack buffer starting at location 0x04FF.