

# Practica No.1 GNU/Linux & Microcontrolador

Departamento Académico de Sistemas Digitales  
Instituto Tecnológico Autónomo de México  
Primavera 2019

JEAN PAUL VIRUEÑA ITURRIAGA  
FABIÁN ORDUÑA FERREIRA

155265  
159001

**Abstract** - Esta práctica es una introducción al Laboratorio de Principios de Mecatrónica en la cual se trabajó con la terminal de GNU/Linux, Git y Arduino para que el equipo tenga un acercamiento más profundo a estas herramientas. Para eso se hicieron ejercicios con los comandos básicos de la terminal y de Git y se realizaron circuitos con una tarjeta Arduino para prender un led en función de ciertas restricciones establecidas en programas precargados a la tarjeta.

## I. INTRODUCCIÓN

La terminal de un sistema operativo es una herramienta poco conocida pero muy poderosa ya que si se sabe utilizar, se puede llegar a trabajar con una computadora de manera rápida y mucho más eficiente al trabajar en un lenguaje de bajo nivel. Por otra parte, Git se ha convertido en una herramienta indispensable para trabajar en proyectos que involucren con código ya que permite tener un control de versiones para poder desarrollar programas con la seguridad de tener un respaldo en la nube. Otra herramienta sumamente útil son los sistemas embebidos ya que con ellos se pueden automatizar un sin número de necesidades y saber usarlos es el primer paso poder satisfacer estas necesidades.

Es por eso que el objetivo de esta práctica es aprender a utilizar la terminal de GNU/Linux, Arduino y Git. Es por eso que durante la práctica el equipo realizó dos ejercicios. El primer ejercicio consiste en acceder a crear y clonar un repositorio desde Github para después realizar una serie de comandos desde la terminal. El segundo ejercicio consistió en codificar y cargar algunos códigos al Arduino y crear circuitos básicos con el fin de que la tarjeta Arduino MEGA 2560 prenda un LED siguiendo ciertos requerimientos.

En cuanto a la organización, este reporte se compone de un marco teórico, donde se presentan la tecnología empleada durante la práctica y se explican los conceptos, seguido del desarrollo de la solución propuesta para llegar al objetivo de la

práctica y una breve justificación de dicha solución. Después se hace un análisis de los resultados obtenidos para terminar con las conclusiones personales y los roles de cada integrante del equipo.

## II. MARCO TEÓRICO

**Arduino MEGA 2560:** La Arduino MEGA 2560 es una tarjeta microcontrolador con 54 pines de entrada y salida digital y 16 de entrada y salida analógica, fue diseñada para realizar proyectos de robótica por Arduino, una plataforma de código abierto que se dedica a fabricar hardware y software [1 y 2].

**Git:** Git es un sistema para gestionar y monitorear contenido de un proyecto al almacenarlo dentro de un directorio que se actualiza al indicarle los cambios añadidos al proyecto [6].

**GitHub:** GitHub es un sistema controlador de versiones dentro de una plataforma desarrollada para almacenar código [7].

**Microcontrolador:** un microcontrolador es una computadora que está dentro de un circuito integrado, dicho circuito integrado forma un sistema embebido. Tiene la característica de contar con memoria, tener entradas y salidas programable y un procesador [5].

**Sistema embebido:** un sistema embebido es una computadora dedicada diseñada para realizar pocas funciones. Mientras que las computadoras de propósito general que fueron concebidas para realizar muchas funciones, los sistemas embebidos pueden ser optimizados en cuanto a costos, desempeño, eficiencia, tamaño entre otras cualidades [4].

**Terminal:** Una terminal es una interfaz no gráfica en la se escriben comandos para darle órdenes a la computadora [3].

### III. DESARROLLO

Como parte del primero de los dos ejercicios realizados, hicimos una copia de un repositorio de Github (clonamos el repositorio) con la finalidad de conocer cómo funciona esta tecnología de control de versiones. Nos familiarizamos con el sistema operativo linux creando directorios y archivos desde la terminal, así como copiando ficheros y observando sus propiedades; además, eliminando archivos y empleando otros comandos en la terminal para reproducir tareas específicas.

Los pasos realizados para este ejercicio los presentamos a continuación.

Para familiarizarnos con Git y Github empleamos los siguientes comandos:

- **git clone https://github.com/FabianOrduna/LabMecatronica** para copiar el repositorio de internet.
- **cd LabMecatronica** para ingresar a la carpeta del repositorio recién clonado.

Para familiarizarnos con la creación y eliminación de ficheros y de archivos empleamos los siguientes pasos:

- Crear ficheros y archivos de la misma forma como se encuentran representados en el diagrama de la figura 1 con los comandos de la figura 2.
- Crear el fichero raíz con el siguiente comando **mkdir Dir\_1**.
- Crear los subficheros con el comando **mkdir Dir\_1/Dir\_11**, y además con **mkdir Dir\_12**.
- Crear sucesivamente los ficheros faltantes del diagrama, de la misma forma en que se crearon los ficheros en los puntos a) y b).
- Crear los archivos dentro de una ruta específica de ficheros. En el caso del arch\_111 del diagrama, se crea con el siguiente comando **vi Dir\_1/Dir\_11/Dir\_111/arch\_111**
- Concluir la creación de los archivos faltantes como se hizo en el inciso anterior.

- Para observar la estructura de un fichero y sus sub ficheros y archivos internos, usamos el comando **ls -aR**. El resultado se puede observar en la figura 3.

Para la eliminación de archivos y ficheros recurrimos al manual de la instrucción para eliminar ficheros, ver figura 2.

- Para eliminar el archivo arch\_12111 usamos la instrucción **rm Dir\_1/Dir\_12/Dir\_1211/Dir\_12111/arch\_12111**
- Para eliminar el fichero **Dir\_1211** usamos la instrucción **rm -r Dir\_1/Dir\_12/Dir\_1211/Dir\_1211**
- Para escribir en archivos con base en despliegue de información de archivos en ficheros, ver figura 4, empleamos el siguiente comando: **cat /opt/arduino-1.8.3/examples/\*//\*.ino > example\_file.txt**.  
Donde **cat** muestra las líneas de los argumentos.  
Donde **>** guarda la información en el archivo puesto a continuación.
- Contar las líneas de un archivo en particular, ver figura 5, con el comando **grep y argumento -wc**

Además se empleó la instrucción **top** para conocer qué procesos corren en ese momento determinado y que recursos de la computadora están utilizando, ver figura 6.

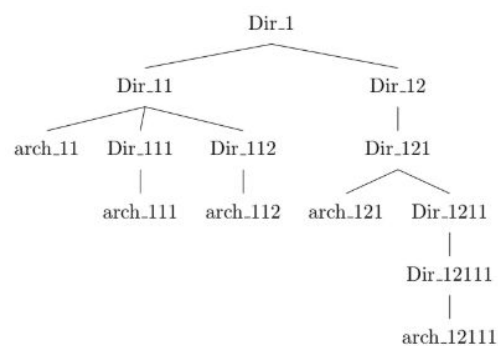


Figura 1. Diagrama de ficheros y archivos.

La segunda parte de los ejercicios consistió en resolver los siguientes problemas empleando un arduino:

1. Simular un alumbrado que responde a cambios en la intensidad luminosa del

ambiente. Mediante una fotoresistencia, se  
sensa la luminosidad del ambiente, al ser  
menor a un valor preestablecido se  
enciende un led. Adicionalmente, el led  
cambia de estado (encendido/apagado) al  
presionar un botón.

2. Realizar un programa que reciba de entrada desde el teclado un número hexadecimal (0-f), valide la entrada y lo despliegue utilizando un display de 7 segmentos.

## IV. RESULTADOS

[illegible]

Figura 2. Instrucciones para crear y eliminar ficheros y archivos.

```
File Edit View Search Terminal Help
root@kali:~/c0131# ./PrinCipalsOfSecuritecnetronics/Practica_15 -a8
..
.. 0x131
./Pr_131
.. 0x131 0x131
./Pr_1/Pr_131
.. arch_11 0x131 0x131
./Pr_1/Pr_131/Pr_131:
.. arch_11
./Pr_1/Pr_131/Pr_131:
.. arch_11
./Pr_1/Pr_131
.. 0x131
./Pr_1/Pr_131/Pr_131:
.. arch_11 0x131 0x131
./Pr_1/Pr_131/Pr_131/Pr_131:
.. arch_11 0x131 0x131
./Pr_1/Pr_131/Pr_131/Pr_131/Pr_131:
.. arch_11 0x131 0x131 0x131 0x131 0x131
```

Figura 3. Resultado del uso del comando `ls -aR` para mostrar estructura interna completa de un fichero.

```

1 #include<stdio.h> //Puls@r402019/PractiseCodeMechElectronics/Practica_15_how_example_file.txt
2
3 //AnalogReadSerial
4 Reads an analog input on pin 0, prints the result to the serial monitor.
5
6 Serial communication is available using serial plotter (Tools -> Serial Plotter menu).
7
8 Attach the center pin of a potentiometer to pin AB, and the outside pins to +5V and ground.
9
10 This example code is in the public domain.
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
101
```

Figura 4. Resultado de aplicar el comando **cat**

```
/opt/arduino-1.8.3/examples/**/*.ino >  
example file.txt
```

```
robotica@labcl03:~/PaulFab2019/PrincipiosDeMecatronica/Practica_1$ grep arduino example_file.txt -wc
```

Figura 5. Resultado de aplicar el contador de líneas al comando empleado en la figura 4.

[illegible]

Figura 6. Procesos y recursos que utilizan de la computadora.

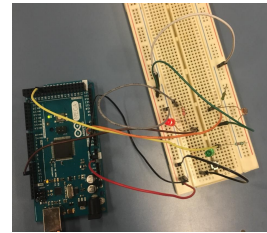


Figura 7. Circuito para resolver problema del botón y de la fotoresistencia mencionados anteriormente.

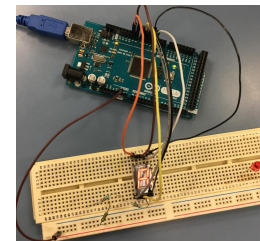


Figura 8. Circuito para resolver problema de entrada por teclado de un número entre 0 y F para desplegar la información al display de 7 segmentos.

## V. CONCLUSIONES

**Jean Paul Virueña Iturriaga**

Fue muy interesante profundizar mi conocimiento sobre la terminal, cada vez le veo más utilidad a la herramienta y me dan ganas de usarla más a menudo. En cuanto a Git, quedé un poco decepcionado de su uso en esta práctica que es una herramienta con la que trabajo frecuentemente. Lo que más me sorprendió fue la simplicidad con la que se trabaja con Arduino.

**Fabián Orduña Ferreira**

Gracias a esta práctica, puede adquirir nuevos conocimientos sumamente relacionados entre sí. En primer lugar, aprendí a manejar la terminal de Linux para crear directorios, archivos, ver la forma en cómo los distintos programas necesitan de cierta capacidad de cómputo; también, logré gracias a

este manejo de la terminal poder trabajar en buena forma con el controlador de versiones Git y con la plataforma de Github de manera básica. Por otro lado puse en práctica los conocimientos teóricos de sistemas embebidos al construir soluciones específicas de los ejercicios planteados a resolver en el Arduino.

## VI. ROLES

A lo largo de la práctica ambos miembros trabajamos en el diseño de las soluciones planteadas. En el reporte Paul se enfocó más a la parte del abstract, la introducción y el marco teórico y Fabián se enfocó más con la parte del desarrollo y de los resultados.

## VII. FUENTES DE CONSULTA

<https://www.techopedia.com/definition/3636/embedded-system>  
<https://www.techopedia.com/definition/3641/microcontroller>  
<https://www.statista.com/statistics/218089/global-market-share-of-windows-7/#0>  
<https://www.techopedia.com/definition/28960/git>  
[https://www.w3schools.com/whatis/whatis\\_github.asp](https://www.w3schools.com/whatis/whatis_github.asp)  
<https://store.arduino.cc/usa/arduino-mega-2560-rev3>  
<https://www.arduino.cc/en/Guide/Introduction>

[1] Arduino. (n.d.). Arduino Mega 2560 Rev3. Retrieved January 31, 2019, from <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

[2] Arduino. (n.d.). What is Arduino? Retrieved January 31, 2019, from <https://www.arduino.cc/en/Guide/Introduction>

[3] EcuRed. (n.d.). Terminal de GNU/Linux. Retrieved January 31, 2019, from [https://www.ecured.cu/Terminal\\_de\\_GNU/Linux](https://www.ecured.cu/Terminal_de_GNU/Linux)

[4] Techopedia. (n.d.). What is an Embedded System? - Definition from Techopedia. Retrieved January 31, 2019, from <https://www.techopedia.com/definition/3636/embedded-system>

[5] Techopedia. (n.d.). What is a Microcontroller? - Definition from Techopedia. Retrieved January 31, 2019, from <https://www.techopedia.com/definition/3641/microcontroller>

[6] Techopedia. (n.d.). What is Git? - Definition from Techopedia. Retrieved January 31, 2019,

from

<https://www.techopedia.com/definition/28960/git>

[7] W3schools. (n.d.). What is GitHub? Retrieved January 31, 2019, from [https://www.w3schools.com/whatis/whatis\\_github.asp](https://www.w3schools.com/whatis/whatis_github.asp)