



Continue

Cours sur l'afficheur 7 segments pdf

Le microcontrôleur pic 16F84A Programmation du PIC 16F84A En général, un afficheur à 7 segments se programme sur 4 bits grâce à 4 entrées Ms.ELN.Smadi%2BBenayallou.pdf type d'afficheurs 7 segments utilisés (anode commune) Ce programme réalise une horloge par affichage multiplexé de 4 afficheurs 7 LIST P = 16C84 LIST kclipac.pdf Positionner le curseur sur PORTB,2 = 1 pour reboucler l'organigramme N'oubliez pas d'enregistrer votre programme Récapitulons l'organigramme : Le port B2 Utilisation logicielCv2.pdf A cet effet, on peut utiliser des afficheurs 7 segments, l'afficheur de caractère à cristaux dans l'afficheur à 7 segments, cet exemple va utiliser le port B du PIC 16F84A Après compilation du programme et simulation sur ISIS, on a un chapitre-5-affichage-des-donnees.pdf Ecrire un programme en C permettant de commander les diodes LED. On désire commander un afficheur à 7 segments afin de réaliser un compteur modulo 10 une LED par plusieurs endroits à l'aide d'un microcontrôleur 16F84A via les td-microcontrôleur.D et des atticheurs 7 segments, gestion d'un clavier, etc L'objectif de ce TP est d'écrire un programme C sous MikroC qui utilise les LEDs et les ONs desirez clignoter un LED à l'aide d'un microcontrôleur 16F84A comme donné par tp_prog_c_emberqu.pdf programme et que vous pensez à le rendre le plus clair possible (ex : en choisissant des . La fenêtre « Logic Analyzer » permet l'affichage des éléments logiques (1 bit) en fonction du « Time Configuration des interruptions, cf sections 9 et enonce_MSIS.pdf 5 1 2 Contrôle de l'affichage dynamique 7 segments Donc, un 16F84-04 est un PIC Mid-Range (16) dont la mémoire programme est de type FLASH (F) 311962715.pdf Tout mécanisme évite de consacrer tous le temps d'exécution du programme à l'affichage et permet de faire d'autres tâches parallèle ? 5 Quelle condition sur TP Atmega32U4_7seg.pdf l'adresse sélectionnée de manière à faciliter l'écriture de segments de programme. Deux dispositifs Figure 2: Organisation de deux dispositifs PIC 16C84 TDUc.pdf Licences CC BY-NC-SA Vous connaissez les afficheurs 7 segments ? fedevofuxoniso.pdf Ou alors vous ne savez pas que ça s'appelle comme ça ? Il s'agit des petites lumières qui forment le chiffre 8 et qui sont de couleur rouge ou verte, la plupart du temps, mais peuvent aussi être bleus, blancs, etc. On en trouve beaucoup dans les radio-réveils, car ils servent principalement à afficher l'heure. Autre particularité, non seulement de pouvoir afficher des chiffres (0 à 9), ils peuvent également afficher certaines lettres de l'alphabet. mario_galaxy_2_iso_it.pdf Matériel Pour ce chapitre, vous aurez besoin de : Un (et plus) afficheur 7 segments (évidemment) 8 résistances de 330Ω330Ω Omega330Ω Un (ou deux) décodeurs 7 segments Une carte Arduino ! Mais dans un premier temps on va d'abord bien saisir le truc avant de faire du code Nous allons commencer par une découverte de l'afficheur, comment il fonctionne et comment le brancher-t-il. Ensuite nous verrons comment l'utiliser avec la carte Arduino. Enfin, le chapitre suivant amènera un TP résistant les différentes parties vues. Prenez appache côté électronique Comme nous l'indique, l'afficheur 7 segments possède 10 segments. Mais un segment c'est quoi au juste ? Et bien c'est une portion de l'afficheur qui est allumée ou éteinte pour utiliser l'afficheur. Cette portion n'est en fait rien d'autre qu'une LED qui est reliée à l'émetteur avec la cathode commune. On débute donc 8 portions comprenant le plan de séparation à part pour faire ça, il ne compte pas en tant que segment. Regardez à quoi ça ressemble : Un afficheur 7 segments (CC-BY-SA, Pengo) Des LED pour les LED Et des LED, il y en a ! Entre 7 et 8 selon les modèles (électronique et circuit d'émission) mais il n'y a pas de différence entre eux. Voici un schéma de l'afficheur 7 segments. Schéma de l'afficheur 7 segments. Le schéma indique les broches de l'afficheur à l'écran. Il y a 14 broches, mais 6 sont reliées entre elles. Selon que cette broche est la cathode ou l'anode on parlera d'afficheur à cathode commune ou... anode commune (vous suivez ?). Dans l'absolu, ils fonctionnent de la même façon, sauf la manière de les brancher diffère (actif sur état bas ou sur état haut). Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal. En mettant une broche de signal à 0, le courant passera et le segment en question s'allumera.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

Le chiffre 10 signifie qu'il possède 10 broches (5 de part et d'autre du boîtier). Voici une représentation de ce dernier (à gauche) : Boîtier du 7 segments - (source: datasheet) Dénomination des segments - (CC-BY-SA, h2q2bob) Voici la signification des différentes broches : LED de la cathode E LED de la cathode D Anode commune des LED LED de la cathode C facultatif le point décimal. LED de la cathode B LED de la cathode A Anode commune des LED LED de la cathode G Pour allumer un segment c'est très simple, il suffit de la relier à la masse ! Nous cherchons à allumer les LED de l'afficheur, il est donc impératif de ne pas oublier les résistances de limitation de courant ! Exemple Pour commencer, vous allez tout d'abord mettre l'afficheur à cheval sur la plaque d'essai (breadboard). Ensuite, trouvez la broche représentant l'anode commune et reliez la à la future colonne du +5V.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

En mettant une broche de signal à 0, le courant passera et le segment en question s'allumera.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

Le chiffre 10 signifie qu'il possède 10 broches (5 de part et d'autre du boîtier). Voici une représentation de ce dernier (à gauche) : Boîtier du 7 segments - (source: datasheet) Dénomination des segments - (CC-BY-SA, h2q2bob) Voici la signification des différentes broches : LED de la cathode E LED de la cathode D Anode commune des LED LED de la cathode C facultatif le point décimal. LED de la cathode B LED de la cathode A Anode commune des LED LED de la cathode G Pour allumer un segment c'est très simple, il suffit de la relier à la masse ! Nous cherchons à allumer les LED de l'afficheur, il est donc impératif de ne pas oublier les résistances de limitation de courant ! Exemple Pour commencer, vous allez tout d'abord mettre l'afficheur à cheval sur la plaque d'essai (breadboard). Ensuite, trouvez la broche représentant l'anode commune et reliez la à la future colonne du +5V.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

En mettant une broche de signal à 0, le courant passera et le segment en question s'allumera.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

Le chiffre 10 signifie qu'il possède 10 broches (5 de part et d'autre du boîtier). Voici une représentation de ce dernier (à gauche) : Boîtier du 7 segments - (source: datasheet) Dénomination des segments - (CC-BY-SA, h2q2bob) Voici la signification des différentes broches : LED de la cathode E LED de la cathode D Anode commune des LED LED de la cathode C facultatif le point décimal. LED de la cathode B LED de la cathode A Anode commune des LED LED de la cathode G Pour allumer un segment c'est très simple, il suffit de la relier à la masse ! Nous cherchons à allumer les LED de l'afficheur, il est donc impératif de ne pas oublier les résistances de limitation de courant ! Exemple Pour commencer, vous allez tout d'abord mettre l'afficheur à cheval sur la plaque d'essai (breadboard). Ensuite, trouvez la broche représentant l'anode commune et reliez la à la future colonne du +5V.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

En mettant une broche de signal à 0, le courant passera et le segment en question s'allumera.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

Le chiffre 10 signifie qu'il possède 10 broches (5 de part et d'autre du boîtier). Voici une représentation de ce dernier (à gauche) : Boîtier du 7 segments - (source: datasheet) Dénomination des segments - (CC-BY-SA, h2q2bob) Voici la signification des différentes broches : LED de la cathode E LED de la cathode D Anode commune des LED LED de la cathode C facultatif le point décimal. LED de la cathode B LED de la cathode A Anode commune des LED LED de la cathode G Pour allumer un segment c'est très simple, il suffit de la relier à la masse ! Nous cherchons à allumer les LED de l'afficheur, il est donc impératif de ne pas oublier les résistances de limitation de courant ! Exemple Pour commencer, vous allez tout d'abord mettre l'afficheur à cheval sur la plaque d'essai (breadboard). Ensuite, trouvez la broche représentant l'anode commune et reliez la à la future colonne du +5V.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

En mettant une broche de signal à 0, le courant passera et le segment en question s'allumera.

Si la broche de signal est à l'état haut, le potentiel est le même de chaque côté de la LED, donc elle est bloquée et ne s'allume pas ! Que l'afficheur soit à anode ou à cathode commune, on doit toujours prendre en compte qu'il faut ajouter une résistance de limitation de courant entre la broche isolée et la broche de signal. Traditionnellement, on prendra une résistance de 330Ω330Ω Omega330Ω pour une tension de +5V, mais cela se calcule (cf. chapitre 1, partie 2). Si vous voulez augmenter la luminosité, il suffit de diminuer cette valeur. Si au contraire vous voulez diminuer la luminosité, augmenter la résistance. Choix de l'afficheur. Pour la rédaction j'ai fait le choix d'utiliser des afficheurs à anode commune et ce n'est pas anodin. En effet et on l'a vu jusqu'à maintenant, sur branches les LED du +5V vers la broche de la carte Arduino. dead led 1 hacked all guns unlocked Ainsi, dans le cas d'un afficheur à anode commune, les LED seront branchées d'un côté au +5V, et de l'autre côté aux broches de signaux. Ainsi, pour allumer un segment est état bas ou sur état haut. Cathode commune ou Anode commune Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal.

Le chiffre 10 signifie qu'il possède 10 broches (5 de part et d'autre du boîtier). Voici une représentation de ce dernier (à gauche) : Boîtier du 7 segments - (source: datasheet) Dénomination des segments - (CC-BY-SA, h2q2bob) Voici la signification des différentes broches : LED de la cathode E LED de la cathode D Anode commune des LED LED de la cathode C facultatif le point décimal. LED de la cathode B LED de la cathode A Anode commune des LED LED de la cathode G Pour allumer un segment c'est très simple, il suffit de la relier à la masse ! Nous cherchons à allumer les LED de l'afficheur, il est donc impératif de ne pas oublier les résistances de limitation de courant ! Exemple Pour commencer, vous allez tout d'abord mettre l'afficheur à cheval sur la plaque d'essai (breadboard). Ensuite, trouvez la broche représentant l'anode commune et reliez la à la future

courant // tant qu'on a pas affiché ce chiffre pendant au moins 500 millisecondes // permet donc de pouvoir lire le nombre affiché while((millis()-temps) < 500) { // on affiche le nombre // d'abord les dizaines pendant 10 ms // le transistor de l'afficheur des dizaines est saturé, // donc l'afficheur est allumé digitalWrite(alim_dizaine, HIGH); // on appelle la fonction qui permet d'afficher le chiffre dizaine afficher(dizaine); // l'autre transistor est bloqué et l'afficheur éteint digitalWrite(alim_unite, LOW); delay(10); // puis les unités pendant 10 ms // on éteint le transistor allumé digitalWrite(alim_dizaine, LOW); // on appelle la fonction qui permet d'afficher le chiffre unite afficher(unite); // et on allume l'autre digitalWrite(alim_unite, HIGH); delay(10); } } // fonction écrivant sur un seul afficheur // on utilise le même principe que vu plus haut void afficher(char chiffre) { digitalWrite(bit_A, LOW); digitalWrite(bit_B, LOW); digitalWrite(bit_C, LOW); digitalWrite(bit_D, LOW); if(chiffre >= 8) { digitalWrite(bit_D, HIGH); chiffre = chiffre - 8; } if(chiffre >= 4) { digitalWrite(bit_C, HIGH); chiffre = chiffre - 4; } if(chiffre >= 2) { digitalWrite(bit_B, HIGH); chiffre = chiffre - 2; } if(chiffre >= 1) { digitalWrite(bit_A, HIGH); chiffre = chiffre - 1; } } Le compteur de 0 à 99 Voilà donc la vidéo présentant le résultat final : Et la même chose sur simulateur interactif. Là encore j'ai du modifier un peu le circuit pour utiliser le CD4511 et donc des afficheurs à cathode commune. !(Contraintes des événements Comme vous l'avez vu juste avant, afficher de manière alternative n'est pas trop difficile. Cependant, vous avez sûrement remarqué, nous avons utilisé des fonctions bloquantes (delay). Si jamais un événement devait arriver pendant ce temps, nous aurions beaucoup de chance de le rater car il pourrait arriver "pendant" un délai d'attente pour l'affichage. Pour parer à cela, je vais maintenant vous expliquer une autre méthode, préférable, pour faire de l'affichage. Elle s'appuiera sur l'utilisation de la fonction millis(), qui nous permettra de générer une boucle de rafraîchissement de l'affichage. Voici un organigramme qui explique le principe : organigramme de rafraîchissement Comme vous pouvez le voir, il n'y a plus de fonction qui "attend". Tout se passe de manière continue, sans qu'il n'y ait jamais de pause.

Ainsi, cet événement ne sera pas (en théorie, un événement très très rapide pourra toujours passer inaperçu). Voici un exemple de programmation de la boucle principale (suite de ses fonctions annexes) : // définition des broches du décodeur 7 segments // vous pouvez changer les numéros si vous voulez const int bit_A = 2; const int bit_B = 3; const int bit_C = 4; const int bit_D = 5; // définition des broches des transistors pour chaque afficheur const int alim_dizaine = 6; // les dizaines const int alim_unite = 7; // les unités // variable pour l'affichage bool afficheur = false; long tempsaffichage = 0; long tempscomptage = 0; int valeur = 0; void setup() { // Les broches sont toutes des sorties pinMode(bit_A, OUTPUT); pinMode(bit_B, OUTPUT); pinMode(bit_C, OUTPUT); pinMode(bit_D, OUTPUT); pinMode(alim_dizaine, OUTPUT); pinMode(alim_unite, OUTPUT); // Les broches sont toutes des sorties l'état bas digitalWrite(bit_A, LOW); digitalWrite(bit_B, LOW); digitalWrite(bit_C, LOW); digitalWrite(bit_D, LOW); digitalWrite(alim_dizaine, LOW); digitalWrite(alim_unite, LOW); } void loop() { // test de rafraîchissement // on fait plus de 10 ms pour afficher un chiffre de 7 segments (alternance unité <-> dizaine) if(chiffre <= 250) { // on inverse la valeur de l'afficheur // pour une unité d'affichage // pour une unité d'affichage (unité ou dizaine) alors que l'afficheur affiche la valeur // on met à jour le temps } // fonction permettant d'afficher un nombre // elle affiche soit les dizaines soit les unités void afficher_nombre(valeur, afficheur), tempsaffichage = millis(); // on met à jour le temps / ici, on peut traiter les événements (button...) // Par exemple on incrément le compteur toutes les secondes if(millis() - tempscomptage > 1000) { valeur++; tempscomptage = millis(); } // on met à jour le temps / on récupère toutes les secondes if(millis() - tempscomptage > 1000) { valeur++; tempscomptage = millis(); } // on récupère toutes les secondes if(millis() - tempscomptage > 1000) { valeur++; tempscomptage = millis(); } // on récupère toutes les secondes if(millis() - tempscomptage > 1000) { valeur++; tempscomptage = millis(); } // on affiche les deux chiffres en utilisant millis Voici l'exemple avec le simulateur (volontairement lent pour voir l'affichage en alternance) : !(Ce chapitre vous a appris à utiliser un nouveau moyen pour afficher des informations avec votre carte Arduino. L'afficheur peut sembler peu utilisé mais en fait de nombreuses applications existe ! (chronomètre, réveil, horloge, compteur de passage, afficheur de score, etc.). Par exemple, il pourra vous servir pour déboguer votre code et afficher la valeur des variables souhaitées... Un simple bouton Sommaire [TP] Parking Academia.edu uses cookies to personalize content, tailor ads and improve the user experience. By using our site, you agree to our collection of information through the use of cookies. To learn more, view our Privacy Policy.